



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY  
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY**

DEPARTMENT OF CONTROL AND INSTRUMENTATION

**ŘÍZENÍ LABORATORNÍHO MODELU  
SYNCHRONNÍCH OS**

CONTROL OF THE LABORATORY MODEL OF SYNCHRONOUS DRIVES

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

Bc. Martin Koubík

**VEDOUCÍ PRÁCE**

SUPERVISOR

prof. Ing. František Zezulka, CSc.

BRNO 2017

## Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**  
Ústav automatizace a měřicí techniky

**Student:** Bc. Martin Koubík

**ID:** 154776

**Ročník:** 2

**Akademický rok:** 2016/17

**NÁZEV TÉMATU:**

### Řízení laboratorního modelu synchronních os

#### POKYNY PRO VYPRACOVÁNÍ:

1. Prostudujte fyzický model synchronních os, řízených čtyřmi synchronními motory a průmyslovým IPC firmy Beckhoff. Důkladně se seznámte s řídicím software TwinCAT pro R – T komunikaci v síti EtherCAT a programovým SW modulem pro programování synchronních motorů.
2. Upravte HW řešení modelu za účelem bezproblémového chodu.
3. Nahradte provizorní ozubená kola konečným řešením s plastovými ozubenými koly.
4. Vypracujte demonstrační program umožňující různé scénáře chodu laboratorního modelu.
5. Odladěný program uložte do paměti řídicího IPC.
6. Přípravek opatřete HW ovládacím panelem za účelem uživatelsky příjemného ovládání laboratorního modelu.
7. Vypracujte podrobný návod pro práci s laboratorním modelem synchronních os.

#### DOPORUČENÁ LITERATURA:

- [1] TwinCAT Manual. Firemní literatura fy Beckhoff  
[2] Koubík Martin, Bc. Semestrální práce. UAMT FEKT 2016

**Termín zadání:** 6. 2. 2017

**Termín odevzdání:** 15.5.2017

**Vedoucí práce:** prof. Ing. František Zezulka, CSc.

**Konzultant:**



doc. Ing. Václav Jirsík, CSc.  
předseda oborové rady



#### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

## **Abstrakt**

Tato práce se zabývá návrhem řídicího programu pro synchronní model os a přestavbou tohoto modelu. První hlavní částí je popis modelu synchronních os. Tedy popis motorů a ostatních prvků modelu. Dále jsou potom probrány všechny úpravy modelu. Další částí je pak probrání programového prostředí TwinCAT a popsání všech důležitých prvků, které je nutno provést pro správnou funkčnost modelu a následný popis funkčních bloků, které jsou použity pro vytvoření programu. Nakonec je popsán celý program.

## **Klíčová slova**

TwinCAT, Synchronní motor, EtherCAT, Industrial PC

## **Abstract**

The aim of this theses describes the design of the control program for the synchronous axes model and reconstruction of this model. The first main part is description of synchronous axes model, description of engines and other components of model. Then model modifications are described. Another part is discussing the TwinCAT programming environment and describe all important elements that must be performed for proper functionality of the model. There is also a description of function blocks that are used to program motion sequences. The last part is described of demonstration program and program which would be used for teaching.

## **Keywords**

TwinCAT, synchronous engine, EtherCAT, Industrial PC

KOUBÍK, M. *Řízení laboratorního modelu synchronních os*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 78 s. Vedoucí diplomové práce prof. Ing. František Zezulka, CSc.

## Prohlášení

„Prohlašuji, že svou diplomovou práci na téma řízení laboratorního modelu synchronních os jsem vypracoval samostatně pod vedením vedoucího semestrální práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. Díl 4 Trestního zákoníku č. 40/2009 Sb.“

V Brně dne: .....

.....

Podpis autora

# Obsah

1	Úvod.....	10
2	Řízení fyzických modelů .....	11
3	Model synchronních os.....	13
3.1	Použité motory .....	14
3.1.1	Definice servomotoru .....	14
3.1.2	ILA2E.....	14
3.2	Použité prvky .....	17
3.2.1	Industrial PC.....	17
3.2.2	Coupler a I/O moduly .....	18
4	Přestavba modelu .....	21
4.1	Komunikační karty EK1122.....	21
4.2	Řešení chyby realizace modelu .....	22
4.3	Konektory RJ45, MOLEX.....	23
4.4	Ovládací panel.....	24
4.4.1	Zapojení ovládacího panelu.....	26
4.4.2	Návod k ovládacímu panelu .....	29
4.5	Ozubená kola.....	30
5	TwinCAT a práce s ním.....	32
5.1	System Manager a jeho nastavení.....	33
5.1.1	Základní konfigurace.....	33
5.1.2	Další nastavení .....	39
5.2	Manuální ovládání os .....	42
5.3	PLC Control .....	43
5.3.1	Spuštění PLC Control.....	44
5.3.2	Příprava knihoven.....	45
5.3.3	Programování .....	46
5.3.4	PLC Control Online režim.....	57
5.3.5	Vlastní program.....	59
5.4	Program pro výuku.....	70
6	Závěr .....	74

# Seznam obrázků

Obr. 1 - Model synchronních os [1] .....	13
Obr. 2 - Rozbor označení motoru [7] .....	15
Obr. 3 - Použité motory [7] .....	15
Obr. 4 - Konektory v elektronice motoru [7] .....	16
Obr. 5 - Použité IPC [2] .....	18
Obr. 6 - Coupler [3] .....	19
Obr. 7 - Schéma komunikace [4] .....	19
Obr. 8 - Jednotlivé moduly [5] .....	20
Obr. 9 - Komunikační karta EK1122 [11] .....	22
Obr. 10 - Zapojení stínění ze strany motoru [7] .....	23
Obr. 11 - Konektor ZS1090-0003 [12] .....	23
Obr. 12 – MOLEX MX-43025-1200 [15] .....	24
Obr. 13 – MOLEX 430300007 [14] .....	24
Obr. 14 - Krabice poskládaná z jednotlivých dílů .....	25
Obr. 15 - Víko pro ovládací panel .....	26
Obr. 16 - Sestava modulů .....	27
Obr. 17 - Zapojení tlačítek .....	27
Obr. 18 - Zapojení kontrol .....	28
Obr. 19 - Zapojení napájení .....	28
Obr. 20 - Zapojení ovládacího panelu .....	29
Obr. 21 - Ovládací panel .....	30
Obr. 22 – Ozubená kola modelu zhotovená 3D tiskem .....	31
Obr. 23 – Nová ozubená kola na modelu .....	31
Obr. 24 - Nastavení IP adresy PC .....	33
Obr. 25 - Výběr připojeného zařízení .....	34
Obr. 26 - Rozšířené hledání připojených zařízení .....	34
Obr. 27 - Akční tlačítka v TwinCATu .....	35
Obr. 28 - I/O konfigurace, Scan Devices .....	35
Obr. 29 - Konfigurace před úpravami modelu .....	36
Obr. 30 - Nová konfigurace (po úpravách modelu) .....	37
Obr. 31 - Připojení motoru k Axis .....	38
Obr. 32 – Seznam Axis .....	38
Obr. 33 - Přidání startup parametrů .....	39

Obr. 34 – Seznam startup parametrů .....	40
Obr. 35 - Okno editace startup parametru, ukázka přidání prvního startup paramteru .....	40
Obr. 36 - Nastavení jednotek a „Display(Only)“ .....	41
Obr. 37 - Nastavení otáček a monitorování .....	41
Obr. 38 - Nastavení Scaling Factor .....	42
Obr. 39 - Manuální ovládání os v TwinCATu .....	43
Obr. 40 - Výběr typu systému .....	44
Obr. 41 - Vytvoření nového POU.....	45
Obr. 42 - Přidání a prohlížení knihoven .....	46
Obr. 43 - Globální proměnné .....	47
Obr. 44 - Funkční blok MC_Power .....	48
Obr. 45 - Funkční blok MC_MoveAbsolute.....	48
Obr. 46 - Funkční blok MC_MoveRelative.....	49
Obr. 47 - Funkční blok MC_MoveVelocity .....	49
Obr. 48 - Funkční blok MC_MoveModulo .....	50
Obr. 49 - Funkční blok MC_Halt .....	50
Obr. 50 - Funkční blok MC_Stop.....	51
Obr. 51 - Funkční blok MC_SetPosition .....	51
Obr. 52 - Funkční blok MC_GearIn .....	52
Obr. 53 - Funkční blok MC_GearOut .....	52
Obr. 54 - Funkční blok MC_Reset .....	52
Obr. 55 - Funkční blok Porovnání.....	53
Obr. 56 - Funkční blok Nerovnost.....	54
Obr. 57 - Vytvoření souboru pro připojení projektu v System Manageru .....	55
Obr. 58 - Připojení PLC programu k projektu v System Manageru.....	55
Obr. 59 - Vstupy a výstupy PLC programu.....	55
Obr. 60 - Linkování proměnné SQ3 na referenční snímač .....	56
Obr. 61 - Linkování Axis .....	57
Obr. 62 - Záložka online a její možnosti (PLC Control) .....	58
Obr. 63 - Boot Settings.....	59
Obr. 64 - Ověření linkovaných proměnných .....	72



## Seznam tabulek

Tab. 1 - Přiřazení pinů konektoru CN4 [7].....	17
Tab. 2 - Přiřazení pinů konektoru ZS1090-0003 [12].....	23
Tab. 3 - Vstupní signály motoru, piny MX-43025-1200 [7].....	24
Tab. 4 - Seznam použitých prvků na ovládacím panelu .....	27
Tab. 5 - Linkování proměnných .....	72

# 1 ÚVOD

Synchronní řízení motorů je vhodné pro použití tam, kde je nutná přesnost. Dostupný model slouží pro ukázkou toho, jak synchronizace motorů funguje. Motory, na kterých jsou ozubená kola, je potřeba úhlově vůči sobě natočit tak, aby roztočená ozubená kola při přiblížení do sebe přesně zapadla.

Toho je docíleno tak, že se motory nejprve natočí do nulové pozice a poté je jedním z nich podle potřeby pootočeno o tolik stupňů, aby došlo ke správnému zapadnutí zubu ozubeného kola jednoho motoru do mezery ozubeného kola druhého motoru. Toto natočení je možné realizovat pomocí funkčního bloku MC\_MoveModulo. Při správném nastavení parametrů motoru může modulo pozice odpovídat stupni natočení tedy  $0^\circ$  -  $360^\circ$ .

Motory sloužící pro lineární pohyb těchto dvou motorů s ozubenými koly je potřeba nejprve dostat do referenční polohy (Obr. 1 zobrazuje model, motory v levé části slouží právě pro pohyb motorů s ozubenými koly). Ta je určena referenčními snímači, na kterých je potřeba motor zastavit programově. Jsou zde ještě další dva krajní snímače. Ty jsou zapojeny do svorek motoru, které slouží jako kontrola limitní polohy. Při njetí na tyto snímače se motory okamžitě zastaví bez ohledu na program. To je z důvodu, aby motory mechanicky nenarážely do gumových zářezek modelu. Jakmile tedy motory dojedou na referenční snímače, je nutné je zde programově zastavit a pro nastavení referenční polohy se jim přepíše jejich absolutní poloha na 0. Pro další pohybování těmito motory se potom použije absolutní polohování s tím, že jeden motor se bude pohybovat do záporných hodnot polohy a druhý do kladných.

## 2 ŘÍZENÍ FYZICKÝCH MODELŮ

Firma Beckhoff implementuje otevřené automatizační systémy založené na PC Control technology. Jejich sortiment zahrnuje industriální PC (dále IPC), I/O komponenty a sběrnice, technologii pohonů a automatizační software. K dispozici mají výrobky pro všechna průmyslová odvětví, které mohou být použity buď jako samostatné komponenty, nebo jako integrované do úplného řídicího systému.

Filozofie Beckhoffu „New Automation Technology“ představuje univerzální a otevřené řešení pro řízení a automatizaci, které se používá na celém světě v široké škále různých aplikací, a to od CNC strojů až po inteligentní automatizaci budov.

Celá firma je rozdělena do čtyř odvětví a to do The IPC Company, The I/O Company, The Motion Company a The Automation Company.

### The IPC Company

Firma Beckhoff nabízí industriální počítač pro každou aplikaci. Jedná se o vysoce kvalitní komponenty založené na otevřených standardech a robustních konstrukcích a krytech. IPC jsou tak ideálně vybaveny na všechny požadavky na řízení. IPC jsou také k dispozici v miniaturním provedení, které je vhodné pro instalaci na DIN lištu. IPC firmy Beckhoff jsou tak vhodné všude tam, kde je vyžadována spolehlivá a robustní PC technologie.

Toto odvětví tedy nabízí následující výrobky:

- Panel PCs (Dotykové panely se zabudovaným IPC)
- Control cabinet PCs (Řídící skříně IPC)
- Control Panel (Dotykový panel bez IPC)
- Embedded PC (Modulární IPC pro instalaci na DIN lištu)
- Motherboards (Industriální základní deska)

### The I/O Company

Firma Beckhoff také nabízí kompletní řadu sběrnicových komponent pro všechny běžné I/O a sběrnicové systémy. U sběrnicových terminálů nabízí třídu krytí IP20 a u boxovaných sběrnicových modulů třídu krytí IP67. Kompletní sortiment je tak k dispozici pro všechny důležité typy signálů a průmyslové sběrnice. Kromě klasických sběrnic nabízí dále EtherCAT sběrnici. Jedná se o vysokorychlostní Ethernetovou sběrnici. Jsou pro ní dostupné EtherCAT terminály, boxy a Plug-in moduly.

Toto odvětví nabízí:

- Bus Terminal (Modulární sběrnice pro automatizaci)
- EtherCAT (Real-time ethernet sběrnice)
- Fieldbus Box (Kompaktní moduly s krytím IP67)

- Lightbus (Rychlé sběrnice s optickými vlákny)
- Infrastructure components (PC sběrnice karty, Switche, EtherCAT komponenty)

### The Motion Company

V kombinaci s řešením Motion Control, které nabízí TwinCAT automatizační software, Beckhoff Drive Technology představuje moderní a kompletní řídicí systém. Řídicí technologie od Beckhoffu založená na PC se ideálně hodí pro jednoosé a víceosé polohování s vysokými dynamickými požadavky.

V tomto odvětví nabízí:

- Servo Drives (Servo pohony jednoosé a víceosé s EtherCAT komunikací nabízejí maximální výkon a dynamiku)
- Synchronous Servomotors (Servomotory s one-cable technologií, která kombinuje napájení a zpětnou vazbu, snižují náklady)
- Linear Servomotors XTS (eXtended Transport System – nahrazuje klasické mechanické systémy za inovativní mechatronické)

### The Automation Company

Firma Beckhoff nabízí komplexní systémová řešení v různých výkonnostních třídách pro všechny oblasti automatizace. Beckhoff řídicí technologie je škálovatelná od vysoce výkonných IPC až po mini PLC a lze ji přesně přizpůsobit dané aplikaci. TwinCAT automatizační software integruje řízení v reálném čase s PLC, NC a CNC funkcemi v jednom balení.

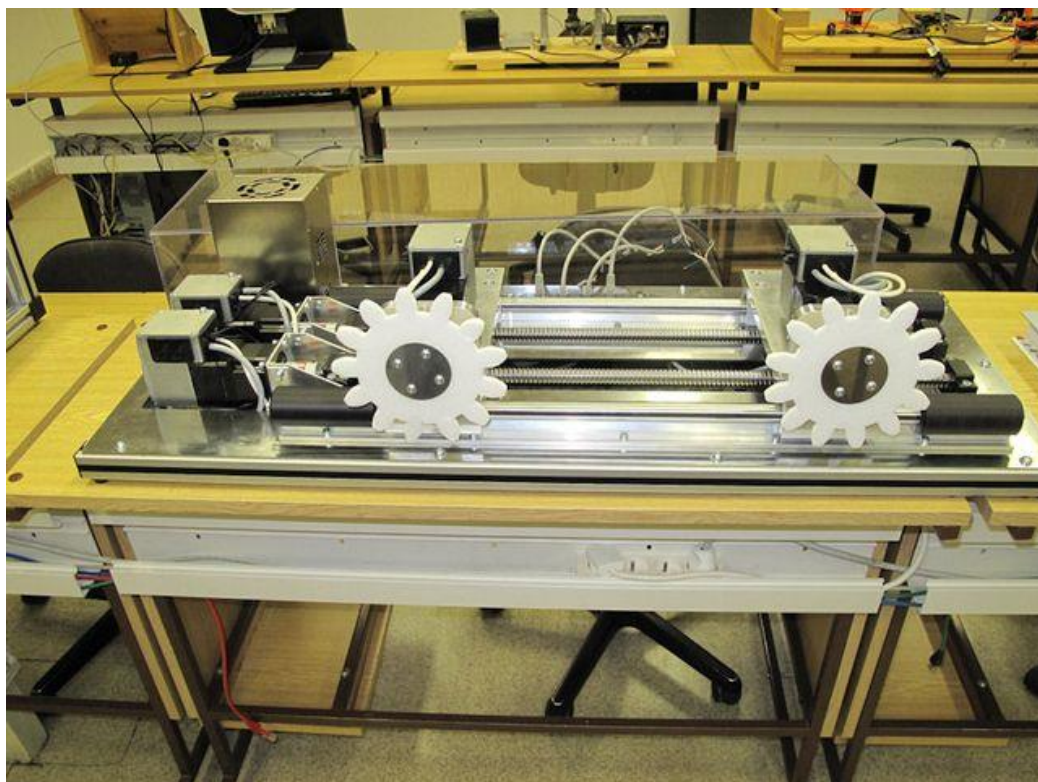
Zde nabízí:

- TwinCAT 3 (Novější programovací prostředí)
- TwinCAT 2 (Starší programovací prostředí)
- TwinSAFE (Otevřená bezpečnostní technologie)

### 3 MODEL SYNCHRONNÍCH OS

Jedná se o model (Obr. 1), který je využíván pro výuku na Vysoké škole polytechnické v Jihlavě. Tento model obsahuje celkem 4 synchronní servomotory ILA2E571TB1A0 od firmy Schneider electric. Dva motory slouží pro rotaci ozubených kol. Tyto dva motory jsou umístěny na dvou kuličkových šroubech motorů, které slouží pro pohyb motorů s ozubenými koly k sobě a od sebe. Všechny motory jsou propojeny pomocí průmyslové sběrnice EtherCAT a jsou napojeny na řídicí kartu, která je dále připojena na industriální PC firmy Beckhoff. Model obsahuje na obou stranách, kam je možný pohyb motorů, vždy dva snímače, kde jeden slouží pro získání referenční polohy a druhý slouží jako koncový (bezpečnostní). Ten slouží k okamžitému zastavení motoru v případě, že se motor na tento snímač dostane. Koncové snímače a referenční snímače jsou zapojeny do motorů do svorek, které slouží jako I/O signály.

Všechny kabely, které se musejí fyzicky pohybovat pro účely lineárního pohybu motorů, jsou vedeny pomocí energetických řetězů, které jsou uspořádány tak, že dovolují pohyb bez zbytečného napínání kabelů. Zároveň poslouží jako ochrana proti mechanickému poškození.



Obr. 1 - Model synchronních os [1]

## 3.1 Použité motory

Jak již bylo popsáno výše, jsou zde použity 4 stejné střídavé synchronní servomotory od firmy Schneider electric. Jedná se o jednofázové synchronní motory s 6 póly. Jelikož v tomto modelu je elektronika motoru napájena 24V, je jejich jmenovitá rychlost 5000 ot/min. Při napájení 48V dosahuje jmenovitá rychlost 7000 ot/min.

### 3.1.1 Definice servomotoru

Servomotor je motor pro elektrické pohony, u kterých lze na rozdíl od běžného motoru nastavit přesnou polohu natočení osy.

### 3.1.2 ILA2E

Tyto motory jsou použity v daném modelu. Jejich přesné označení je ILA2E 571TB1A0. Pomocí manuálu od výrobce lze z tohoto označení získat informace o daném motoru (Obr. 2). Jedná se tedy o následující parametry:

- ILA = servomotor (motor)
- 2 = 24 ... 48 V<sub>DC</sub> (Napájecí napětí)
- E = EtherCAT (Komunikační rozhraní)
- 57 = 57mm (Velikost)
- 1 = 1 stack (Délka motoru)
- T = Vysoká rychlost rotace/nízký moment (Typ vinutí)
- B = Plošný konektor (na plošném spoji)
- 1 = Singleturn kodér
- A = Bez zádržné brzdy
- 0 = Bez převodovky

	ILA	2	E	57	1	P	B	1	A	0	--
<b>Motor</b> ILA = Servo motor											
<b>Supply voltage</b> 2 = 24 ... 48 V <sub>dc</sub>											
<b>Communication interface</b> E = EtherCAT											
<b>Size</b> 57 = 57 mm											
<b>Length</b> 1 = 1 stack 2 = 2 stacks											
<b>Winding</b> P = Medium speed of rotation/medium torque T = High speed of rotation/low torque											
<b>Connection version</b> B = Printed circuit board connector C = Industrial connector											
<b>Position capture</b> 1 = Singleturn encoder 2 = Multiturn encoder <sup>1)</sup>											
<b>Holding brake</b> A = Without holding brake F = With holding brake <sup>2)</sup>											
<b>Gearbox</b> 0 = Without gearbox											
<b>Reserved</b>											

Obr. 2 - Rozbor označení motoru [7]

Nejedná se přímo o servomotor, ale o zařízení, které obsahuje jak servomotor, tak i řízení, které je v tomto zařízení integrované.



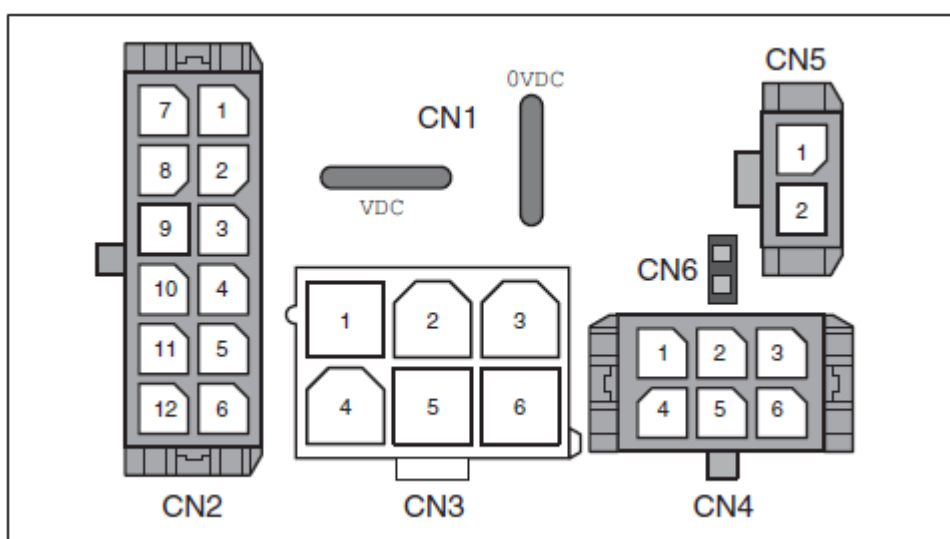
Obr. 3 - Použité motory [7]

## Komponenty zařízení

- Motor (Bezkartáčový střídavý synchronní servomotor s 6 póly)
- Elektronika (řídící elektronika, lze ovládat pomocí komunikačního rozhraní, dále k dispozici 4 digitální 24V signály použitelné jako I/O)
- Kodér (Vybaven jedno-otáčkovým kodérem, vnitřní rozlišení 32768 kroků na otáčku)
- Zadržná brzda (Zařízení jí může být volitelně vybavené. Je řízena automaticky)

## Technická data

Motory, které jsou v tomto modelu, mají v elektronice následující konektory:



Obr. 4 - Konektory v elektronice motoru [7]

- CN1 (Napájecí napětí)
- CN2 (EtherCat signály – galvanicky oddělené)
- CN3 (RS485 signály)
- CN4 (Vstupní a výstupní 24V signály)
- CN5 (Bezpečnostní funkce STO)
- CN6 (Jumper pro zakázání bezpečnostní funkce STO)

### Bezpečnostní funkce STO

Motor je vybaven integrovanou bezpečnostní funkcí STO (Safe Torque Off) (IEC 61800-5-2), která umožňuje pro kategorii 0 zastavení jako IEC 60204-1 bez externího stykače. Není tak nutné pro tuto kategorii přerušit napájecí napětí pro zastavení motoru. Ač je tato funkce k dispozici, není využita a je vypnuta.



## Vstupní a výstupní signály

Motor nabízí možnost připojit až 4 signály do I/O konektoru CN4.

**Tab. 1 - Přiřazení pinů konektoru CN4 [7]**

Pin	Signal	Factory setting
1	+ 24V	
2	I/O 3	Input free available
3	I/O 1	Input positive limit switch (LIMP)
4	0V	
5	I/O4	Input reference switch (REF)
6	I/O2	Input negative limit switch (LIMN)

Signály konektoru CN4 mají továrně přednastavené jisté funkce (Tab. 1). Jediný volný vstup je I/O 3, který je továrně nastaven jako vstupní signál.

Referenční snímače motorů lineárního pohybu jsou připojeny na pin 5 konektoru CN4. Koncové snímače jsou potom zapojeny na pin 3. Jsou tedy napojeny na kladný vstupní limit. V případě, že se tedy motor dostane na tento snímač, dojde k jeho zastavení. Záporný limit v tomto modelu není používán. Kdyby byl využíván, nesměl by v cestě pohybu být druhý motor.

## **3.2 Použité prvky**

Pro řízení modelu jsou použity prvky od firmy Beckhoff, a to hlavně z důvodu real-time řízení pomocí EtherCAT sběrnice.

### **3.2.1 Industrial PC**

Použité IPC je IPC C6915-0000. Je vyobrazeno na Obr. 5. Tento model má následující konfiguraci:

- Procesor Intel Atom Z510 (1,1 GHz)
- RAM 1GB DDR2, rozšiřitelné na 2GB
- Grafický adaptér Intel GMA 500, DVI-D konektor
- Duální Ethernetový adaptér s 2x 10/100/1000BASE-T konektor
- 128 MB Compact Flash karta
- 4x USB 2.0
- Napájecí napětí 24V<sub>DC</sub>
- Operační systém Microsoft Windows CE 6

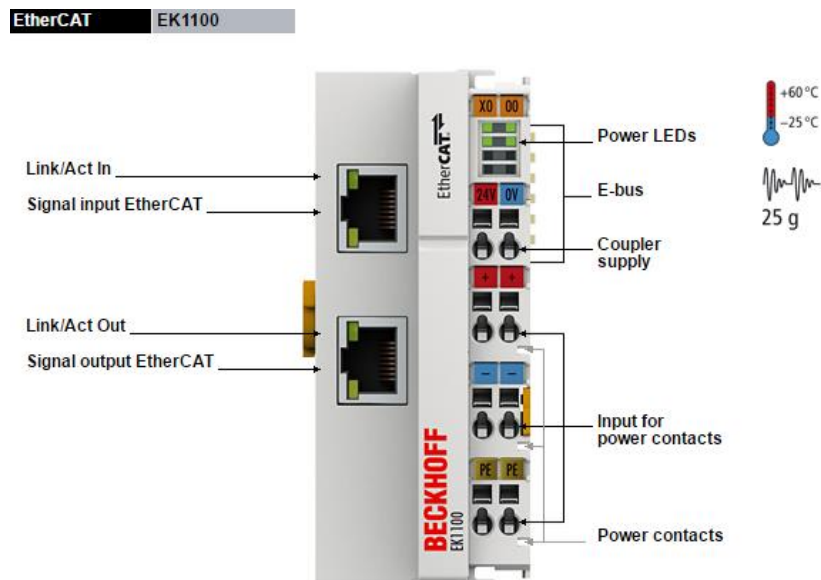


Obr. 5 - Použité IPC [2]

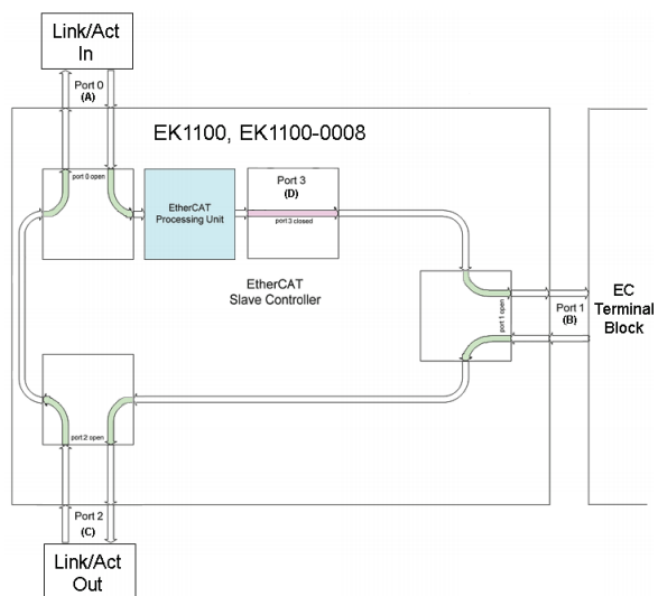
### 3.2.2 Coupler a I/O moduly

Pro komunikaci s motory je v modelu použit coupler (vazební člen) EK1100 (Obr. 6). Ten je zde právě hlavně z důvodu komunikace s motory. Jinak samozřejmě slouží i k připojení různých karet (např. I/O moduly). Možné fyzické topologie EtherCAT jsou hvězda, linie nebo strom. Všechny topologie však tvoří logický kruh, jelikož rámec odeslaný řídicím zařízením prochází přes všechny připojené slave zařízení a posledním je odeslán zpět [16]. Coupler má vstupní a výstupní ethernetový konektor pro síť EtherCAT.

Na Obr. 7 lze vidět, jak je tedy udělané propojení v coupleru. Do vstupního EtherCAT konektoru tak připojíme naše IPC a do výstupního jsou pak připojeny motory.



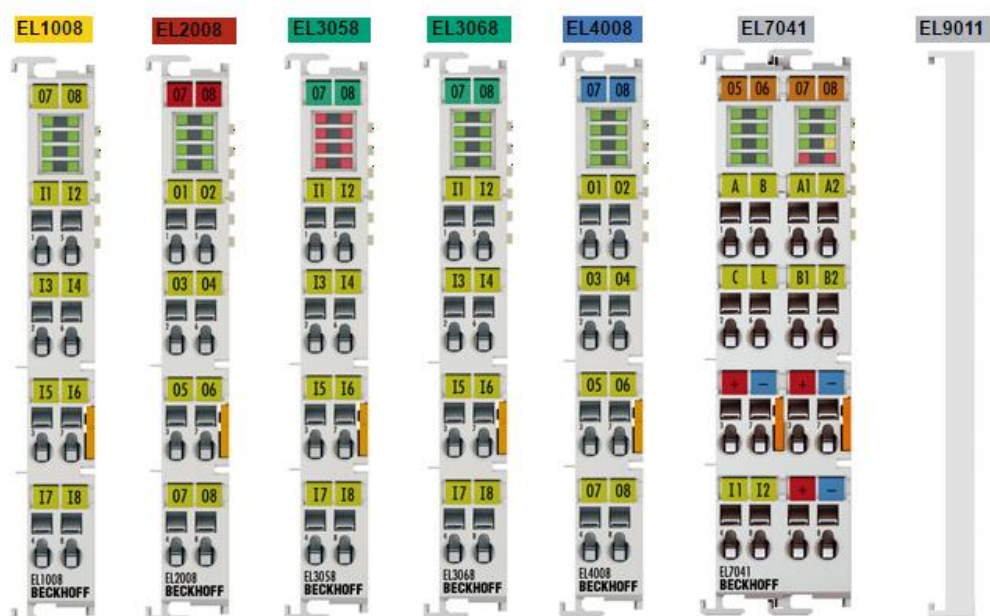
Obr. 6 - Coupler [3]



Obr. 7 - Schéma komunikace [4]

Dále jsou tedy do coupleru připojeny bus terminály, neboli různé moduly, karty (Obr. 8). Ty však nejsou v tomto modelu využity. Náš model obsahuje:

- Modul s digitálními vstupy 2x EL1008(8-Channel, 24V)
- Modul s digitálními výstupy 2x EL2008 (8-Channel, 24V)
- Modul s analogovým vstupem 1x EL3058 (8-Channel, 4-20mA), 1x EL3068(8-Channel, 0-10V)
- Modul s analogovým výstupem 1x EL4008 (8-Channel, 0-10V)
- Modul pro krokové motory 1x EL7041 (1-Channel, 50V, 5A)
- Ukončovací modul 1x EL9011



Obr. 8 - Jednotlivé moduly [5]

## 4 PŘESTAVBA MODELU

Původní model synchronních os je popsán v kapitole výše. Rozhodnutí pro přestavbu modelu vzniklo z důvodu špatné, nestabilní komunikace motorů v síti EtherCat. Původní zapojení komunikačního rozhraní motorů bylo zapojeno do topologie fyzického kruhu. Ač je to jedna z možných topologií sítě, ukázalo se, že pro dané motory to není nejlepší volba. Dalším a nejspíše hlavním prvkem nestabilní komunikace byly nestíněné komunikační kabely motorů.

Úkolem tedy bylo navrhnout řešení, které by bylo funkční a proveditelné.

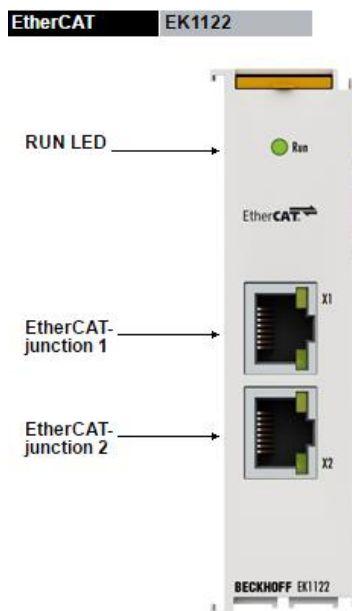
Dále pak bylo úkolem vytvořit jednoduchý ovládací panel, pomocí kterého by bylo možné ovládat, resp. pouštět jednotlivé naprogramované sekvence pohybů modelu synchronních os.

V závěrečné fázi bylo úkolem vyměnit dosud používaná polystyrenová ozubená kola za kola plastová.

### 4.1 Komunikační karty EK1122

Pro zajištění správné funkčnosti a komunikace motorů byly zvoleny komunikační karty EK1122 viz Obr. 9. Tyto karty umožňují zapojení jakéhokoliv zařízení, určeného do sítě EtherCat, prostřednictvím komunikačního kabelu a konektorů RJ45. V podstatě rozšiřují aktuální topologii sítě o topologii hvězda. V našem případě nabízejí možnost pro připojení motorů jednotlivě.

Umožňují připojení až dvou zařízení, přičemž karta sama pozná, zda je v konektorech zapojené nějaké zařízení a podle toho zpřístupňuje jednotlivé porty. Celkově má tedy karta 4 porty. Jeden je jako vstupní, který propojuje komunikaci mezi kartami připojenými před naší kartou, další je brán jako výstupní, je tedy možnost pokračovat dalšími kartami a poslední dva porty potom slouží právě pro připojení dvou zařízení pomocí konektorů RJ45. Při správné konfiguraci karet je správná funkčnost vyjadřována LED diodou umístěnou v horní části karty.



Obr. 9 - Komunikační karta EK1122 [11]

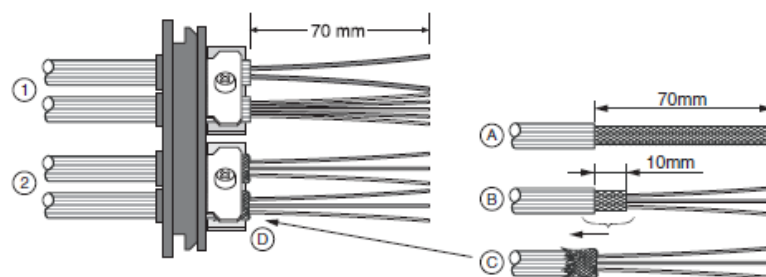
## 4.2 Řešení chyby realizace modelu

Jedním z hlavních důvodů, proč docházelo k výpadkům komunikace s motory, byly nestíněné komunikační kabely, které z důvodu velmi úzkého energetického řetězu vedly v podstatě v těsné blízkosti s napájecími kabely motoru. Docházelo tak k rušením komunikace. To bylo pozorovatelné v System Manageru, kde vždy po čase došlo ke ztrátě spojení s motory. Poté následovalo několik pokusů o obnovení komunikace. Pokud se obnovení podařilo, opět došlo po nějakém časovém intervalu ke ztrátě. V některých případech se komunikace obnovit nepodařila a bylo potřeba odpojit model z napájení a znovu připojit. Ani to však nezaručilo obnovení komunikace s motory.

Možnost vést tyto kabely odděleně od napájecích kabelů motoru není, proto bylo nutné zvolit vhodný kabel pro toto zařízení. Jelikož průmyslové kabely přímo pro EtherCAT jsou drahé, zvolil jsem levnější variantu a to Ethernetový kabel SF/UTP CAT.5e, 4x2xAWG26/7. Jedná se o kabel, který je stíněný fólií a opletením. Jednotlivé žíly kabelu jsou potom lankové z důvodu namáhání na ohyb v energetických řetězech.

O zapojení stínění ze strany komunikační karty se v podstatě stará konektor. Ze strany motoru je potom potřeba stínění přichytit k mechanickým svorkám motoru zabraňujícím nenásilnému pohybu kabelů v motoru. Princip zapojení stínění ze strany motoru lze vidět na Obr. 10.

Po této úpravě a zapojení motorů do karet EK1122 již nenastal problém s komunikací s motory.



Obr. 10 - Zapojení stínění ze strany motoru [7]

### 4.3 Konektory RJ45, MOLEX

Zvolené komunikační karty mají možnost připojit zařízení přes konektory typu RJ45. Firma Beckhoff nabízí robustnější typy konektorů ZS1090-0003 (Obr. 11), které jsou 4 pinové a tudíž nemůže dojít k problému umístění vodiče na jiný pin, než který je používáný.

Tab. 2 - Přiřazení pinů konektoru ZS1090-0003 [12]

	Wire colour			
Function/ Signal	EtherCat	EIA/TIA 568 A	EIA/TIA 568 B	Pin No.
TD+	YE	WH GN	WH OG	1
TD-	OG	GN	OG	2
RD+	WH	WH OG	WH GN	3
RD-	BU	OG	GN	6

Tyto konektory jsou navrženy spíše pro vodiče typu drát, jelikož dochází k zaříznutí nožů, napojených na piny konektoru, do jednotlivých vodičů přes jejich izolaci. Pro náš účel je však požadován vodič typu lanko. Při zařezávání vodičů je tak lanko více ohebné a bylo nutné provést zaříznutí vícekrát za sebou, aby došlo ke správnému narušení izolantu vodiče a tím ke správnému připojení.



Obr. 11 - Konektor ZS1090-0003 [12]

Spolu s konektorem je k dispozici návod, jak správně postupovat při kompletaci konektoru.

Zapojení komunikačního kabelu ze strany motoru je řešeno 12 pinovými konektory MOLEX MX-43025-1200 (Obr. 12), do kterých se jednotlivé vodiče zapojují pomocí kontaktů Micro-Fit 3.0 s označením 430300007 (Obr. 13).



Obr. 12 – MOLEX MX-43025-1200 [15]



Obr. 13 – MOLEX 430300007 [14]

Původní zapojení používalo jak vstupní, tak i výstupní komunikační signály jednotlivých motorů. To bylo z důvodu zapojení motorů za sebou do fyzického kruhu. Pro správné připojení ke komunikačním kartám jsme použili vstupní komunikační signály motoru. V Tab. 3 lze vidět, o jaké piny a signály se jedná.

Tab. 3 - Vstupní signály motoru, piny MX-43025-1200 [7]

Pin	Signál	Barva kabelu
9	Tx+	White/Green
10	Tx-	Green
11	Rx+	White/Orange
12	Rx-	Orange

## 4.4 Ovládací panel

Jedním z úkolů bylo vytvořit jednoduchý ovládací panel za účelem snadného ovládání modelu synchronních os. Prvotní plán byl umístit všechny tlačítka vedle sebe a kontrolky pod ně. Z důvodu omezení 3D tiskárny na velikost tisku 200 mm x200 mm nebyl tento plán realizovatelný. Navrhl jsem tedy panel s 8 ovládacími prvky, které jsou umístěny do dvouřadu 4 tlačítka v jedné řadě, 4 tlačítka v druhé.



Návrh krabičky proběhl pomocí internetové aplikace TinkerCAD[13]. Krabička byla vytištěna na 3D tiskárně. Víko krabičky (Obr. 15) bylo vytištěno v celku, zatímco krabička byla vytištěna po částech z důvodu výšky krabičky a tenkosti stěny. Jelikož 3D tiskárna do výšky materiál pouze vrství, stává se tak tato plocha ne příliš pevná. Část, která se tiskne jako velká plocha, je při 3D tisku tištěna pomocí algoritmů, které vytvářejí pevnější strukturu tištěné plochy. Z tohoto důvodu jsem zvolil možnost, při které se všechny části krabičky vytiskly naležato a po vytištění se slepily dohromady. Náznak toho, jak je krabička slepena, lze vidět na Obr. 14. Veškeré návrhy jednotlivých dílů krabičky lze najít v příloze 1.

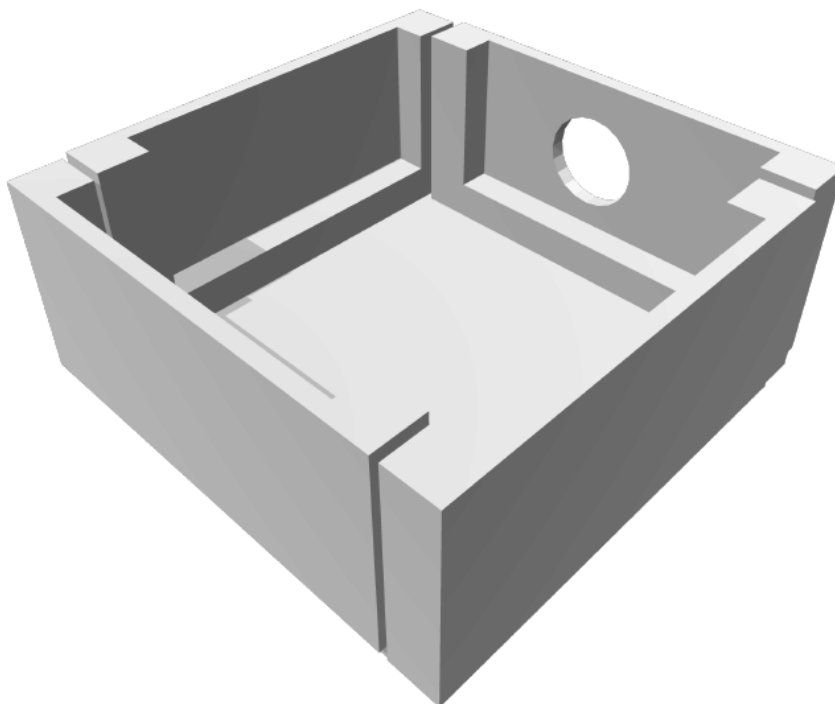
Panel je osazen 7 kontrolkami pro indikaci různých stavů. LED diody, které jsou použity v kontrolkách, jsou dimenzovány na proud 20mA červená a 35 mA zelené. Pro 35 mA použiji k LED diodě odpor 680  $\Omega$  (4.1) a pro 20 mA použiji paralelně řazené odpory 3k3 a 2k. Odpor na červené LED tak bude 1320  $\Omega$ , tudíž proud pro tuto LED bude přibližně 18 mA.

$$I = \frac{U}{R} = \frac{24}{680} = 35\text{mA}$$

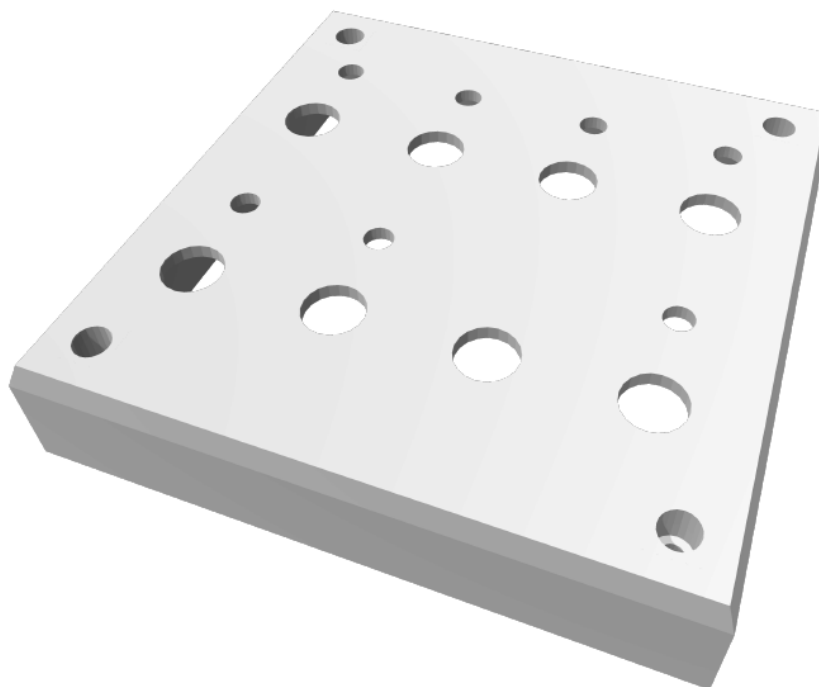
(4.1)

Kde I je proud tekoucí LED, U napětí z výstupní karty a R zvolený rezistor.

Dále je potom osazen 8 tlačítka. Červené tlačítko slouží jako STOP tlačítko pro zastavení modelu, zelené jako START, modré pro inicializaci modelu a ostatní bílá tlačítka jsou jednotlivé programy.



Obr. 14 - Krabička poskládaná z jednotlivých dílů



Obr. 15 - Víko pro ovládací panel

#### 4.4.1 Zapojení ovládacího panelu

Prvky z ovládacího panelu (Tab. 4) jsou zapojeny na vstupní a výstupní digitální karty. Model obsahuje 2 vstupní a 2 výstupní karty zapojené za sebou. Využil jsem vždy první z těchto dvou karet.

Na vstupní kartě jsou připojena tlačítka. Vstupní karta funguje na principu přivedení napětí. Při stisku tlačítka se tedy přivede na vstup napětí, které je bráno jak log. 1. Co se týče výstupní karty, tak na té jsou připojeny kontrolky. Jakmile je jeden výstup aktivní, objeví se na něm 24V. Toto napětí prochází přes příslušný rezistor a kontrolku na 0V. To způsobí rozsvícení kontrolky připojené na příslušném výstupu.

Kontrolky obsahují LED diody, které jsou určeny na určité proudy, proto je potřeba k nim dát do série rezistor o příslušné velikosti pro vytvoření požadovaného proudu. To je řešeno u rovnice 4.1.

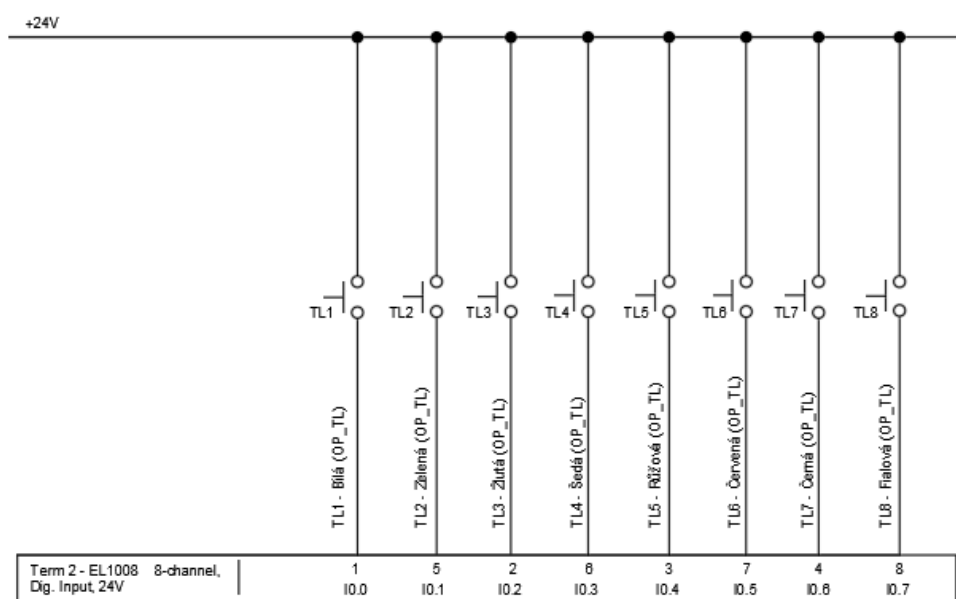
Na Obr. 16 lze vidět kompletní sestavu karet, modulů připojených na coupler. Na Obr. 17 lze vidět zapojení tlačítek na digitální vstupy, na Obr. 18 zapojení LED kontrolky na digitální výstupy a na Obr. 19 zapojení napájení. Na Obr. 20 lze pak vidět zapojení uvnitř ovládacího panelu.

**Tab. 4 - Seznam použitých prvků na ovládacím panelu**

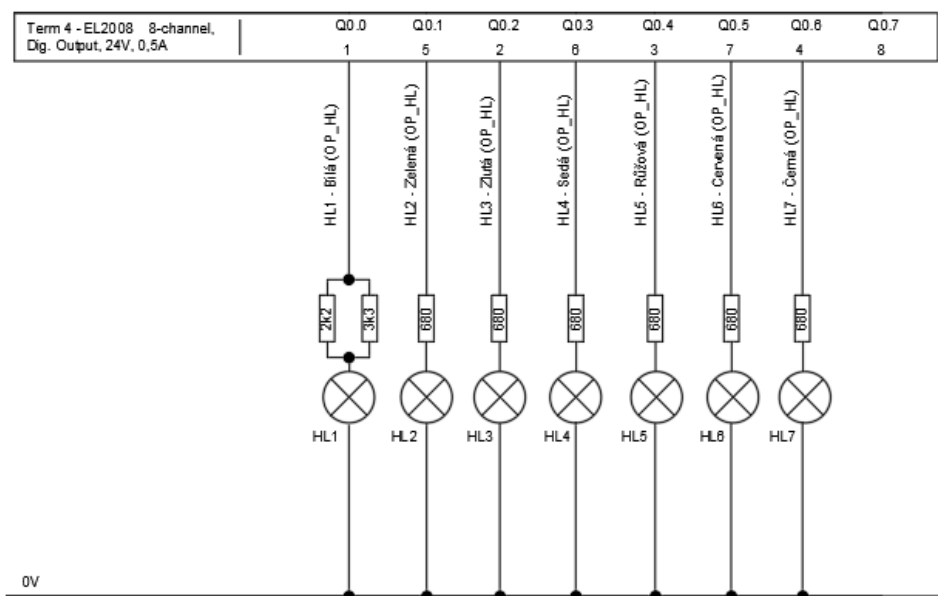
Prvek	Označení	Parametry	Počet
Zelená kontrolka	L-R732G	35mA	6x
Červená kontrolka	L-R732R	20mA	1x
Zelené tlačítko	P-PB303B GREEN	250V, 1A	1x
Modré tlačítko	P-PB303B BLUE	250V, 1A	1x
Červené tlačítko	P-PB303B RED	250V, 1A	1x
Bílé tlačítko	P-PB303B WHITE	250V, 1A	5x

Term 1	Term 2	Term 3	Term 4	Term 5	Term 6	Term 7	Term 8	Term 9	Term 21	Term 25	
BECKHOFF EtherCAT EK1100	Digital Input, 8-channel, 24V  BECKHOFF EL1008	Digital Input, 8-channel, 24V  BECKHOFF EL1008	Digital Output, 8-channel, 24V  BECKHOFF EL2008	Digital Output, 8-channel, 24V  BECKHOFF EL2008	Analog Input, 8-channel, 4-20mA  BECKHOFF EL3068	Analog Input, 8-channel, 0-10V  BECKHOFF EL3068	Analog Output, 8-channel, 0-10V  BECKHOFF EL3048	Stepper motor terminal, 50V DC, 5A, incremental encoder  BECKHOFF EL7041	2-port EtherCAT junction  BECKHOFF EK1122	2-port EtherCAT junction  BECKHOFF EK1122	End cap BECKHOFF EL9011

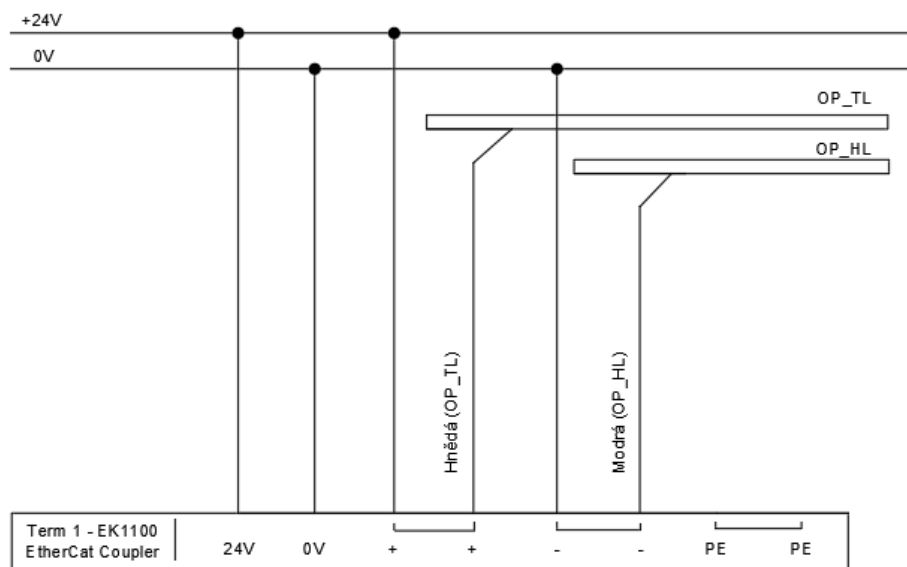
**Obr. 16 - Sestava modulů**



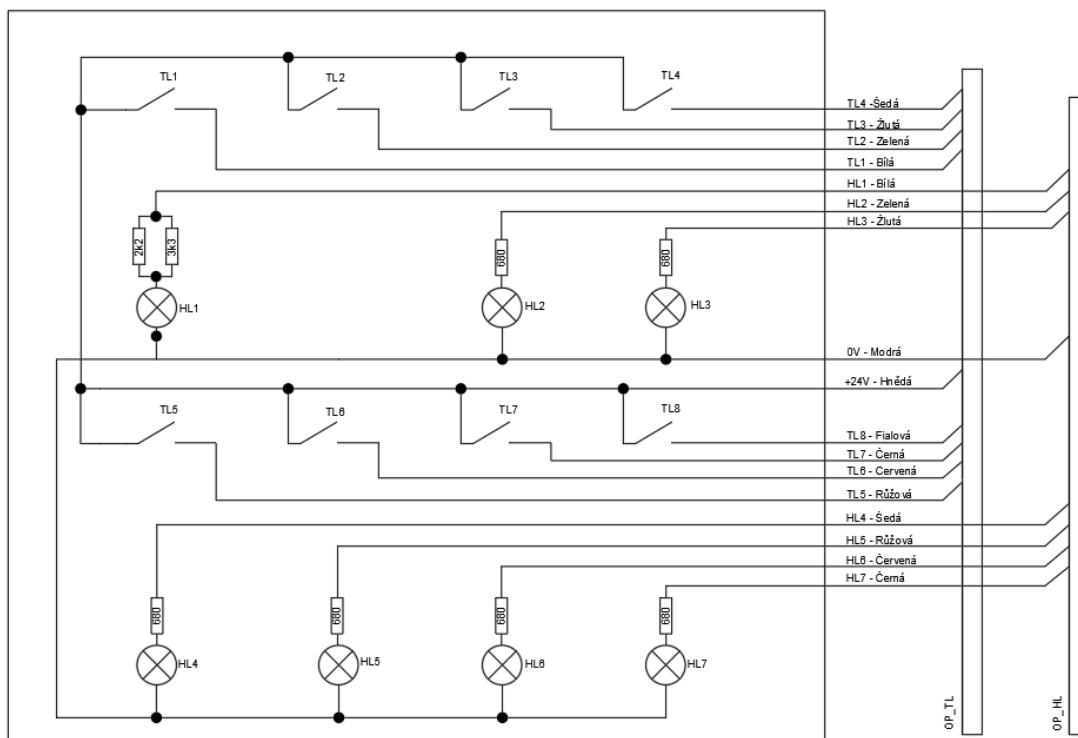
**Obr. 17 - Zapojení tlačítek**



**Obr. 18 - Zapojení kontrolky**



**Obr. 19 - Zapojení napájení**

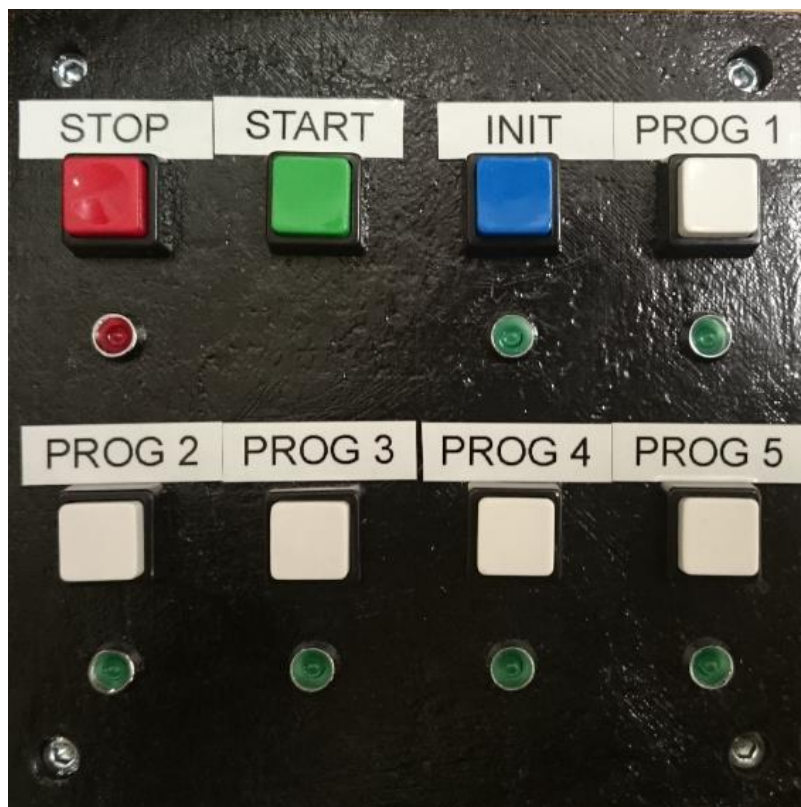


Obr. 20 - Zapojení ovládacího panelu

#### 4.4.2 Návod k ovládacímu panelu

Na Obr. 21 lze vidět rozmístění jednotlivých ovládacích prvků na ovládacím panelu.

Při zapojení modelu do napájení a po zprovoznění IPC, tedy po naběhnutí systému, je třeba stisknout tlačítko START. Dojde tak k zapnutí všech motorů. Poté je již možné stisknout tlačítko INIT pro spuštění první inicializace modelu. Jakmile dojde k dokončení inicializace, rozsvítí se kontrolka pod tlačítkem INIT informující o dokončení inicializace. Následně již lze zvolit jeden z programů. Volbou jednoho z pěti programů se rozsvítí příslušná kontrolka informující o volbě programu. Zvolený program se spustí stiskem tlačítka START. Jakmile se program dokončí, zhasne kontrolka daného programu a spustí se inicializace. Jakmile se rozsvítí kontrolka inicializace, je možné znovu vybrat program a spustit ho. V případě stisknutí tlačítka STOP se celý model zastaví a motory se vypnou. Pro návrat ze stavu STOP je nutné stisknout tlačítko INIT a následně tlačítko START. V případě zvolení více programů najednou je potřeba stisknout tlačítko STOP a poté se z tohoto stavu dostat viz výše.



Obr. 21 - Ovládací panel

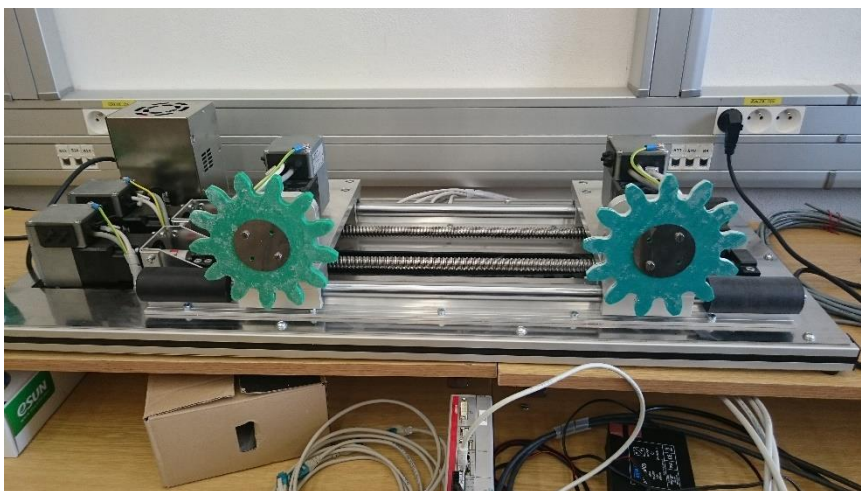
## 4.5 Ozubená kola

Všechny testy byly prováděny s polystyrenovými ozubenými koly z důvodu větší bezpečnosti v případě špatného natočení ozubených kol vůči sobě. Mohlo by tak při splynutí dojít ke zlomení zubů. S těmito zuby se nedalo přibližovat k sobě příliš z důvodu většího zubu než mezery. Docházelo by tak ke tření.

Navrhl jsem tedy, jak by měla ozubená kola vypadat a ta byla vytištěna na 3D tiskárně. Nová ozubená kola (Obr. 22) tak mají o trochu větší mezery než zuby, aby bylo možné zajet s nimi hlouběji do sebe. Rozmístění konstrukčních děr pro upevnění na motory je pak stejné jako u polystyrenových kol. To je hlavně z důvodu, aby při jejich výměně nebylo potřeba přepisovat parametry inicializace při natáčení kol do správné polohy. Nakonec se však konstrukční díry trochu posunuly a ve výsledku bylo potřeba upravit inicializaci. Jedno ozubené kolo tak zůstalo na nulové pozici, zatímco druhé je pootočené o  $15,8^\circ$ . To zajistí bezproblémové zapadnutí zubů do sebe.



**Obr. 22 – Ozubená kola modelu zhotovená 3D tiskem**



**Obr. 23 – Nová ozubená kola na modelu**

## 5 TWINCAT A PRÁCE S NÍM

TwinCat (The Windows Control Automation Technology) je nástroj sloužící k realizaci automatizačních systémů na základě PC techniky. Jedná se tak o vývojový nástroj umožňující změnit téměř jakoukoliv výpočetní sestavu na bázi PC v systém pro řízení v reálném čase obsahující prostředí pro běh programů pro PLC, NC PTP (řízení pohybu os z bodu do bodu), NCi (řízení pohybu os použitím interpolace), plnohodnotné řízení CNC nebo řízení robotů. [6]

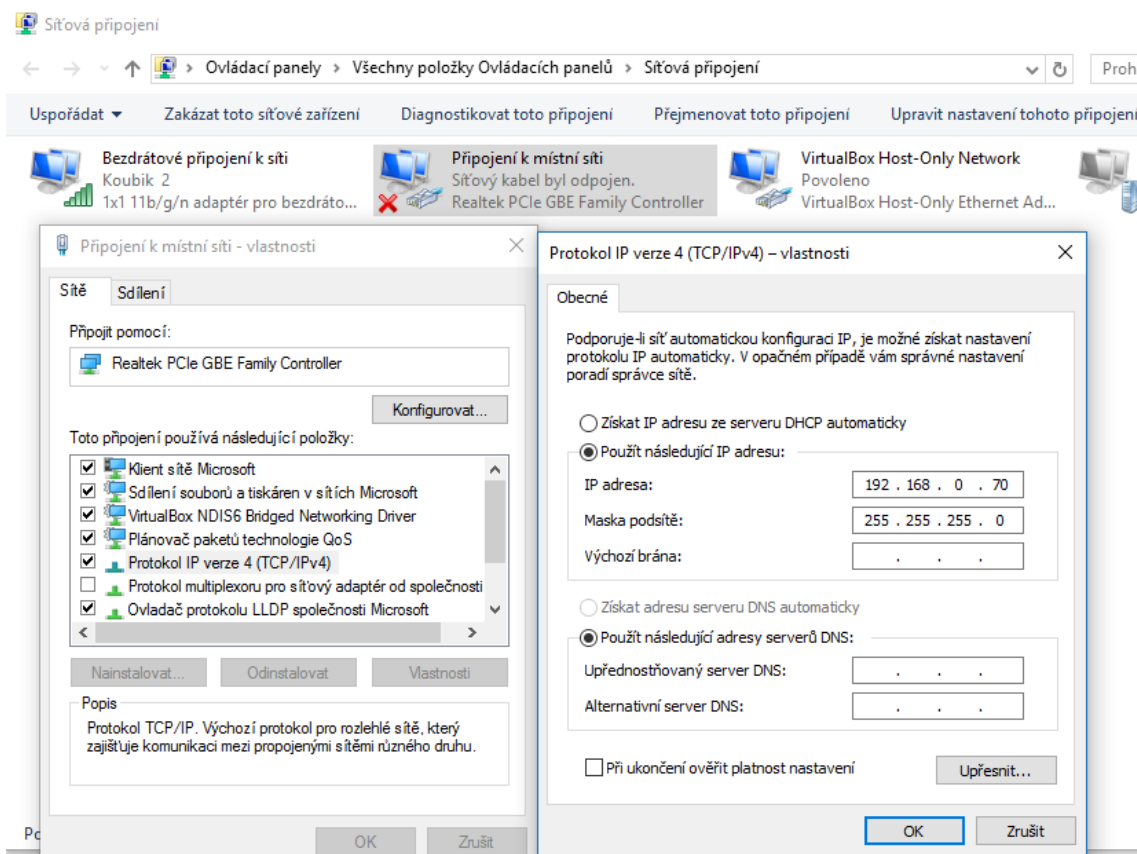
Zdarma jsou dostupné PLC a NC motion control funkce.

V této kapitole je popsáno, jak pracovat s prostředím TwinCAT a jak vše nastavit tak, aby se došlo ke správné funkčnosti modelu synchronních os.

Nejprve je nutné si zjistit, jakou IP adresu má IPC, na který jsme připojeni. Jelikož je možné k němu připojit periferie jako myš, obrazovku, klávesnici, můžeme jednoduše v pracovní liště systému Windows CE, který se nachází na daném IPC, zjistit, jakou IP adresu má. Tvar IP adresy našeho IPC je 192.168.0.X. Tuto adresu si musíme napevno nastavit v PC, pomocí kterého se budeme připojovat k IPC. Na místo X v IP adrese si nastavíme číslo, které se neshoduje s tím, které má naše IPC. IP adresa nastavená na našem PC pak může vypadat následovně: 192.168.0.70.

Na Obr. 24 lze vidět, jak se v PC dostat do nastavení IP adresy. Nejprve se dostaneme přes ovládací panely do síťových připojení. Vybereme síťový adaptér, do kterého se později připojíme s IPC. Na Obr. 24 můžeme vidět, že zvolený adaptér má odpojen síťový kabel. To však ničemu nevadí. Víme, že později se do něj připojíme s naším IPC. Klikneme na něj pravým tlačítkem a zvolíme vlastnosti. Zobrazí se nám tak další okno, ve kterém je rolovací menu. V něm nalezneme položku „Protokol IP verze 4 (TCP/IPv4)“. Zvolíme jej a kliknutím na tlačítko vlastnosti se nám otevře jeho nastavení. Do kolonky IP adresa potom vyplníme požadovanou adresu. Masku podsítě se stiskem tlačítka „tab“ na klávesnici vyplní sama.





Obr. 24 - Nastavení IP adresy PC

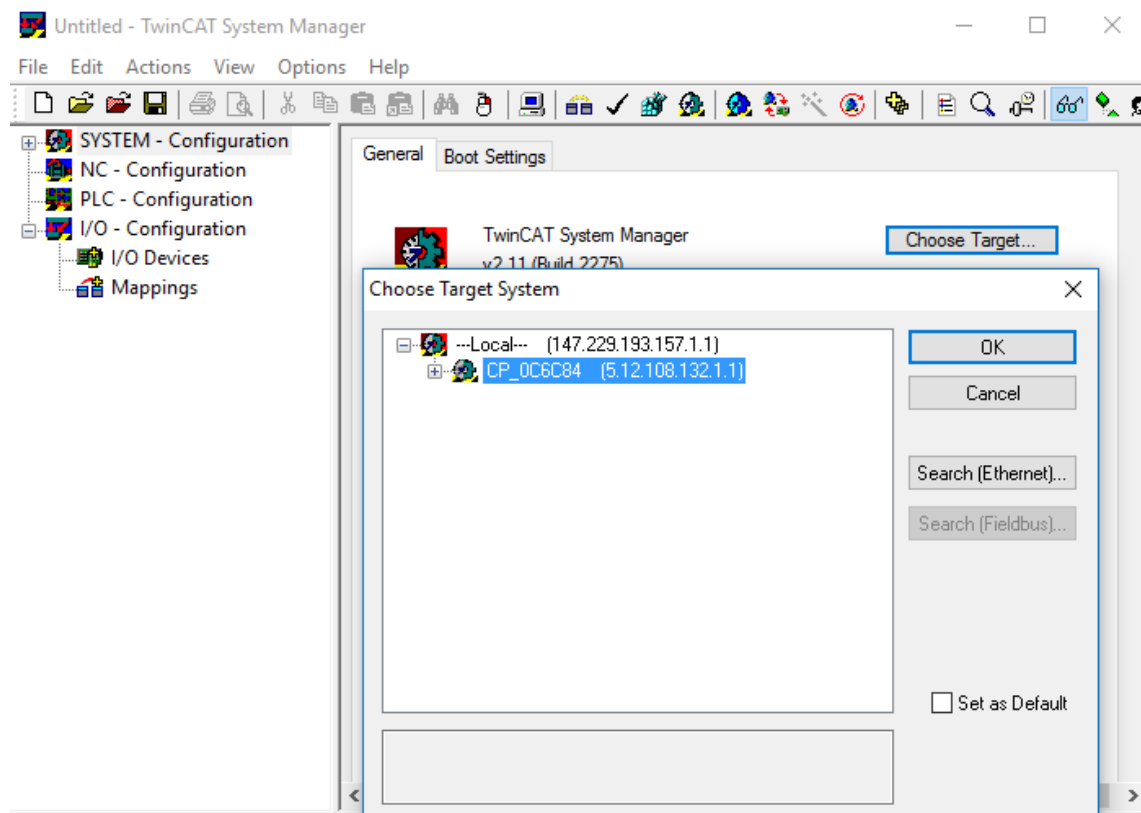
## 5.1 System Manager a jeho nastavení

TwinCAT System Manager je centrální konfigurační nástroj. Je to místo, kde jsou spravovány vstupy a výstupy jednotlivých softwarových úkolů a fyzické vstupy a výstupy připojených sběrnic.

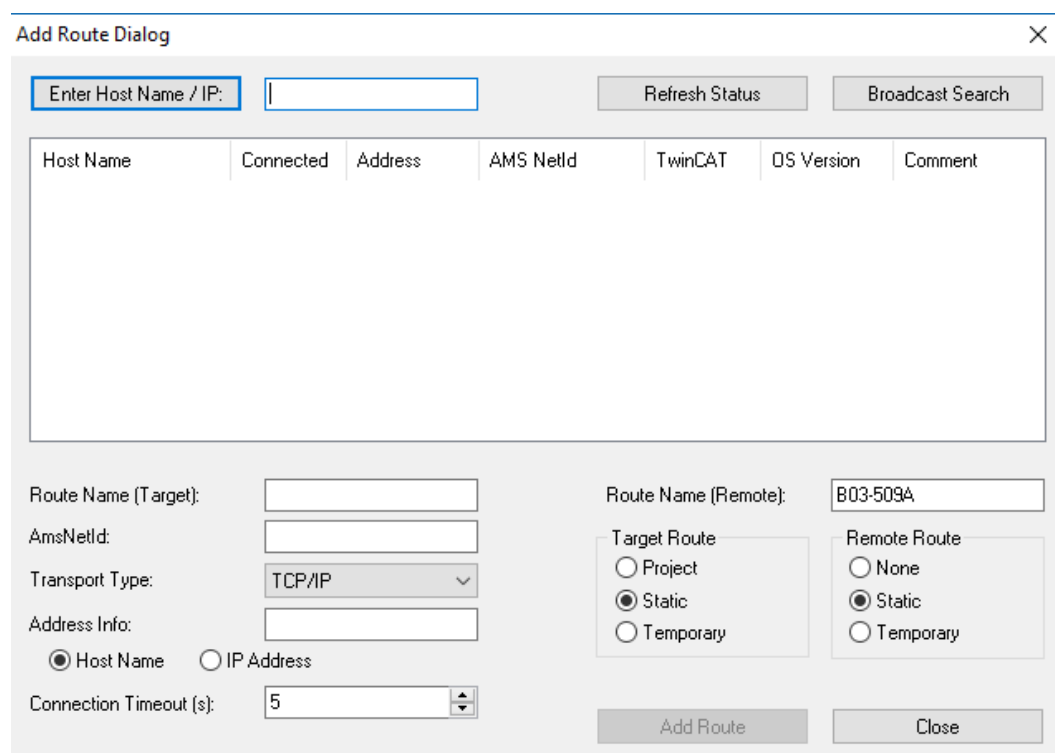
Podporuje všechny standardní komerční komunikační sběrnice (např. EtherCAT, PROFIBUS DP, ProfiNET, CANopen, DeviceNet a další) a některé další standardní PC rozhraní.

### 5.1.1 Základní konfigurace

Při prvním spuštění TwinCAT System Manageru se nám vytvoří nový konfigurační soubor, který si můžeme uložit a nadále s ním pracovat. Označíme si položku „SYSTEM – Configuration“ a dále v pravé části prostředí vybereme záložku „Version (Local)“. Kliknutím na tlačítko „Choose Target...“ (Obr. 25) vybereme, k čemu se chceme připojit (naše IPC), a pokud zde není nic na vybrání, musíme přejít do vyhledávání pomocí tlačítka „Search (Ethernet)“ (Obr. 26). Zde je potom možnost vyhledat zařízení podle jeho IP adresy, kterou již známe z předchozího hledání IP adresy.



Obr. 25 - Výběr připojeného zařízení



Obr. 26 - Rozšířené hledání připojených zařízení

Na Obr. 27 lze vidět tlačítka sloužící pro nastavování různých módů TwinCATu. Tlačítko 1 generuje mapování. Tlačítko 2 slouží pro kontrolu správnosti konfigurace (v podstatě pro kontrolu toho, zda se něco nezměnilo). Tlačítko 3 je potom pro aktivování nastavené konfigurace. Tlačítko 4 slouží pro nastavení nebo pro resetování „Run“ módu. Tlačítko 5 pak slouží pro nastavení nebo pro resetování „Config“ módu.

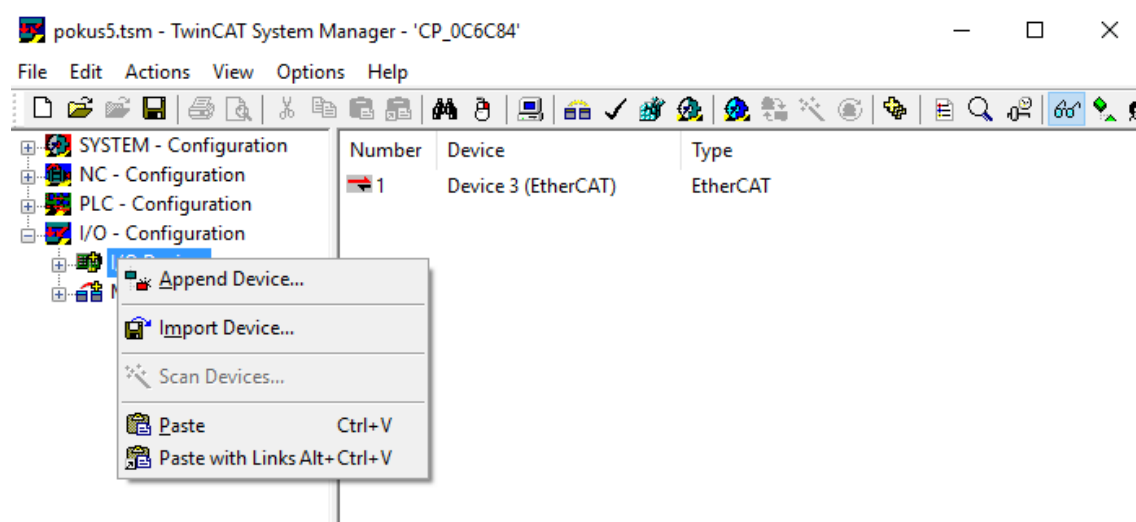


Obr. 27 - Akční tlačítka v TwinCATu

V „Run“ módu musíme být v případě, chceme-li dostávat aktuální informace z připojeného zařízení, nebo používat manuální ovládání jednotlivých os.

V „Config“ módu pak musíme být v případě, že chceme upravovat jakékoliv parametry v System Manageru.

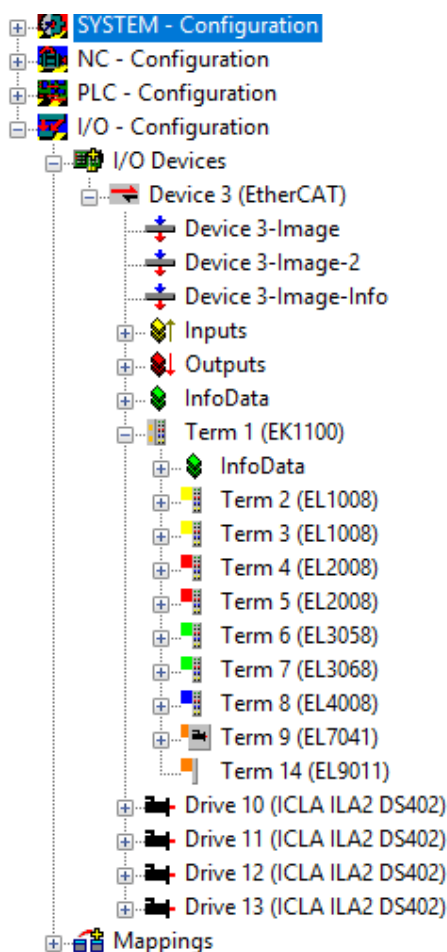
Jakmile se nám povede připojit k zařízení, můžeme přejít ke konfiguraci vstupů a výstupů. Pro tuto činnost musíme být v Config módu. Rozbalíme položku „I/O - Configuration“ a zde vybereme položku „I/O Devices“. Na ní klikneme pravým tlačítkem myši a vybereme „Scan Devices...“ (Obr. 28). System Manager potom prohledá připojená zařízení. Jelikož je na coupler připojen modul pro krokový motor, TwinCat se nás zeptá, zda ho chceme připojit k Axis. Tuto akci potvrdíme.



Obr. 28 - I/O konfigurace, Scan Devices

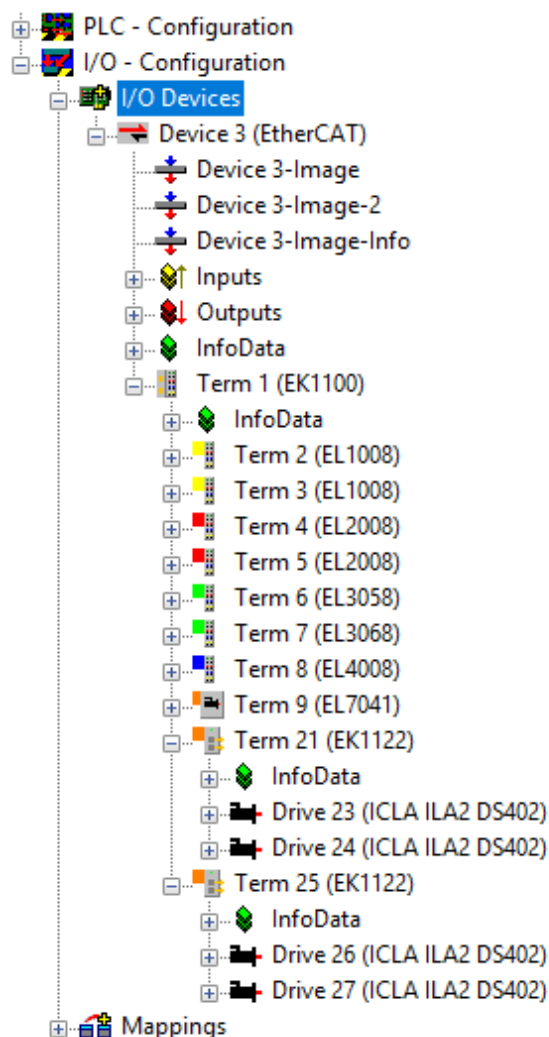
Na Obr. 29 můžete vidět, jak vypadala konfigurace před zásahem do modelu. Motory byly připojené pomocí komunikačního kabelu na výstupním konektoru coupleru. Lze tak pozorovat změny, kdy předtím byly motory mimo coupler EK1100,

tudíž zapojené až za ním, zatím co nyní, viz Obr. 30, jsou motory zapojeny na komunikačních kartách EK1122, které jsou zapojeny za moduly připojenými na coupleru.



**Obr. 29 - Konfigurace před úpravami modelu**

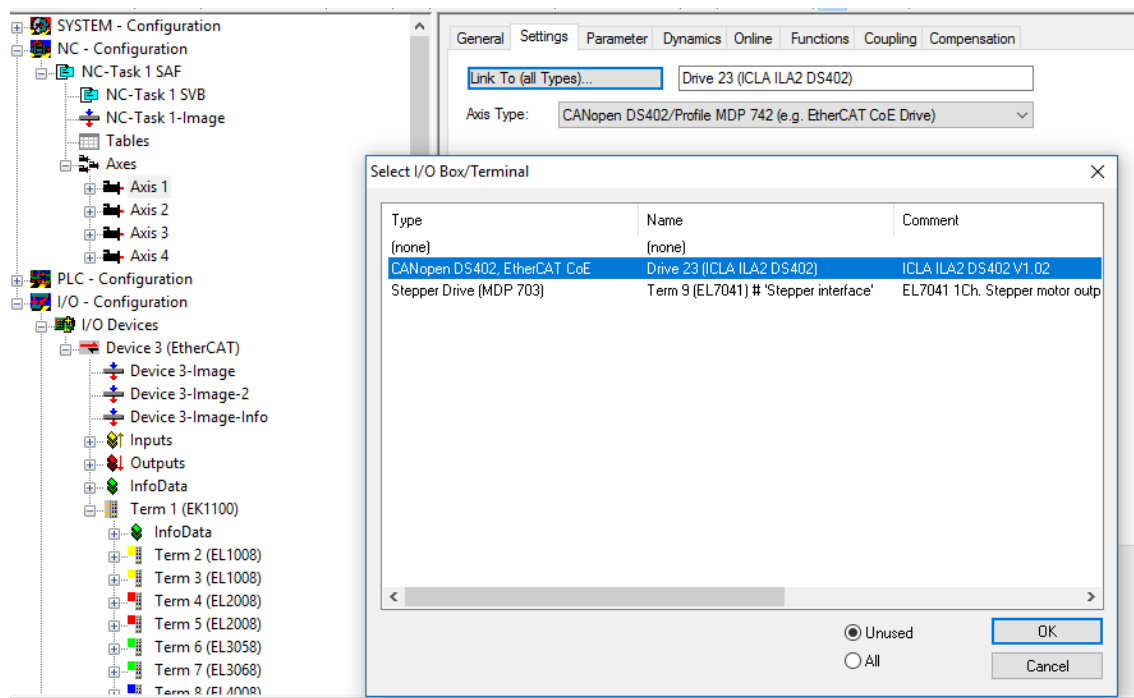
S tím přichází i změna ohledně automatického připojení motorů k Axis. Zatímco u staré konfigurace se motory připojily automaticky k Axis, resp. Axis se samy vytvořily, v nynější konfiguraci je potřeba si Axis vytvořit samostatně a motory k nim připojit.



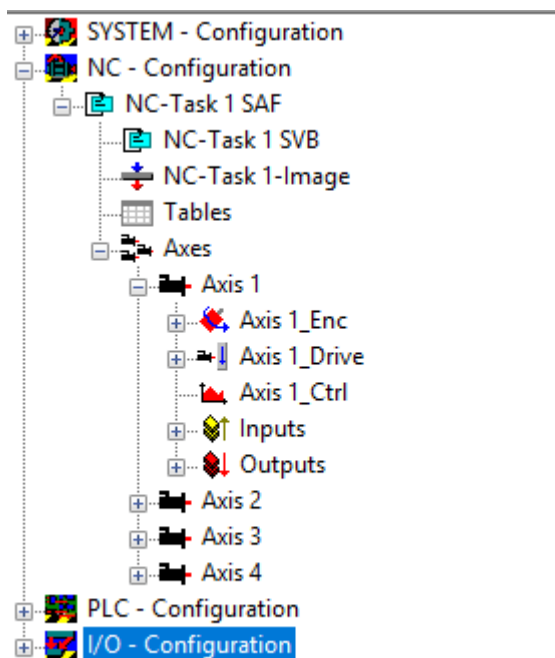
**Obr. 30 - Nová konfigurace (po úpravách modelu)**

Klikneme tedy na „NC-Configuration“. Můžeme vidět, že je zde již vytvořená jistá úloha. Ta se nám vytvořila, pokud jsme potvrdili její vytvoření na základě vytváření konfigurace. Pokud zde však žádná úloha není, klikneme na „NC-Configuration“ pravým tlačítkem a zvolíme „Append Task...“. Když rozbalíme vytvořenou úlohu, najdeme zde položku „Axes“ (Obr. 32). Nalezneme v ní jednu vytvořenou Axis1 pro krokový motor. Tu smažeme, jelikož jí nebudeme potřebovat. Klikneme na položku „Axes“ pravým tlačítkem a zvolíme „Append Axis...“. To nám přidá novou Axis do této úlohy. Provedeme to ještě 3x, abychom měli připravené 4 Axis, pro každý motor jednu. Měli bychom mít vytvořené Axis1 až Axis4. Ke každé ose nyní musíme připojit motor. Pro snadnou orientaci si na Axis1 připojíme první motor, v našem případě Drive 23, na Axis2 Drive 24 atd. Připojení se provede tak, že klikneme na vybranou Axis. Otevře se nám okno v pravé části prostředí TwinCat.

Vybereme záložku „Settings“ a hned jako první tlačítko je „Link to (all Types)...“. Klikneme na něj a v následujícím okně zvolíme jeden z motorů (Obr. 31).



Obr. 31 - Připojení motoru k Axis



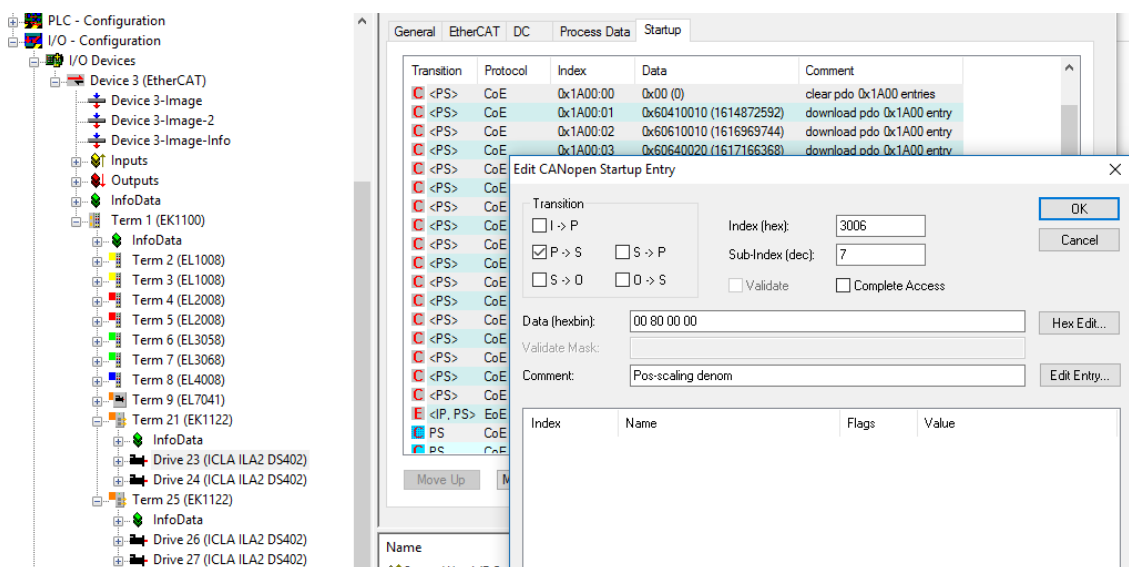
Obr. 32 – Seznam Axis

## 5.1.2 Další nastavení

Máme nyní připravený projekt a můžeme začít konfigurovat jednotlivé motory a osy k nim přiřazené.

### Startup parametry

Pro funkčnost motorů je nejprve nutné jim přidat Start-up parametry, které potřebují pro svůj správný chod. To se udělá v položce „I/O-Configuration“, kde si zvolíme „I/O Devices“ a zvolíme „Device“, které je připojené pomocí EtherCAT v našem případě „Device 3“. To opět rozklikneme a uvidíme coupler EK1100. Ten rozklikneme a uvidíme, co vše je na něm připojeno. Jsou zde i dvě přidané karty EK1122. Každou z nich rozklikneme a uvidíme naše motory Drive 23, Drive 24, Drive 26, Drive 27. Kliknutím vždy na jeden z nich se nám v pravé části prostředí otevře okno s několika záložkami. Zvolíme záložku Startup a klikneme na tlačítko „New“. Otevře se nám okno pro přidání Startup parametru (Obr. 33).



Obr. 33 - Přidání startup parametrů

Postupně přidáváme všechny parametry uvedené na Obr. 34. Po přidání by se měly zobrazit v tabulce, která je v záložce Startup vidět. Stejně postupujeme pro všechny ostatní zařízení Drive. Abychom nemuseli všechny tyto parametry opisovat stále dokola, můžeme si je přidat v jednom zařízení a kliknutím pravým tlačítkem postupně můžeme tyto přidané parametry pomocí „Export to XML“ vyexportovat do námi zvolené složky. U dalších zařízení pak už jen klikneme do tabulky pravým tlačítkem myši a zvolíme „Import from XML“. Vybereme složku, kam jsme si tyto parametry uložili, jeden z nich označíme a dáme přidat. Přidají se nám automaticky všechny parametry ve složce.

PS	CoE	0x3006:07	0x00008000 (32768)	Pos-scaling denom
PS	CoE	0x3006:08	0x00000001 (1)	Pos-scaling num
PS	CoE	0x605A:00	0xFFFE (65534)	Quickstop option code
PS	CoE	0x3006:06	0x0002 (2)	Limit switch behavior
PS	CoE	0x6060:00	0x0008 (8)	Op-mode CSP

Obr. 34 – Seznam startup parametrů

Na Obr. 35 lze vidět, jak vypadá okno editace (přidání) startup parametru.

Obr. 35 - Okno editace startup parametru, ukázka přidání prvního startup paramteru

### Parametry Axis

Dalším nastavováním už nastavujeme jednotlivé parametry os „Axis“. Tyto parametry jsou vždy stejné pro dvojice os (Axis1 a Axis2, Axis3 a Axis4) Uvedeme si příklad nastavení parametrů pro „Axis 1“ a pro „Axis 3“.

Axis (3,4) - Tyto dvě osy by při správném nastavení měly být motory s ozubenými koly. Kliknutím přímo na jednu z těchto os se nám opět v pravé straně zobrazí záložky, u kterých můžeme dále nastavovat parametry. V záložce „Settings“ si nastavíme jednotky „Unit“ na ° (stupně). V části „Display(only)“ zaškrtneme modulo (Obr. 36). V záložce „Parameter“ v kolonce „Velocities“ si můžeme nastavit referenční a maximální otáčky a manuální otáčky. V kolonce „Monitoring“ pak musíme nastavit hodnotu „Position Lag Monitoring“ na FALSE (Obr. 37).

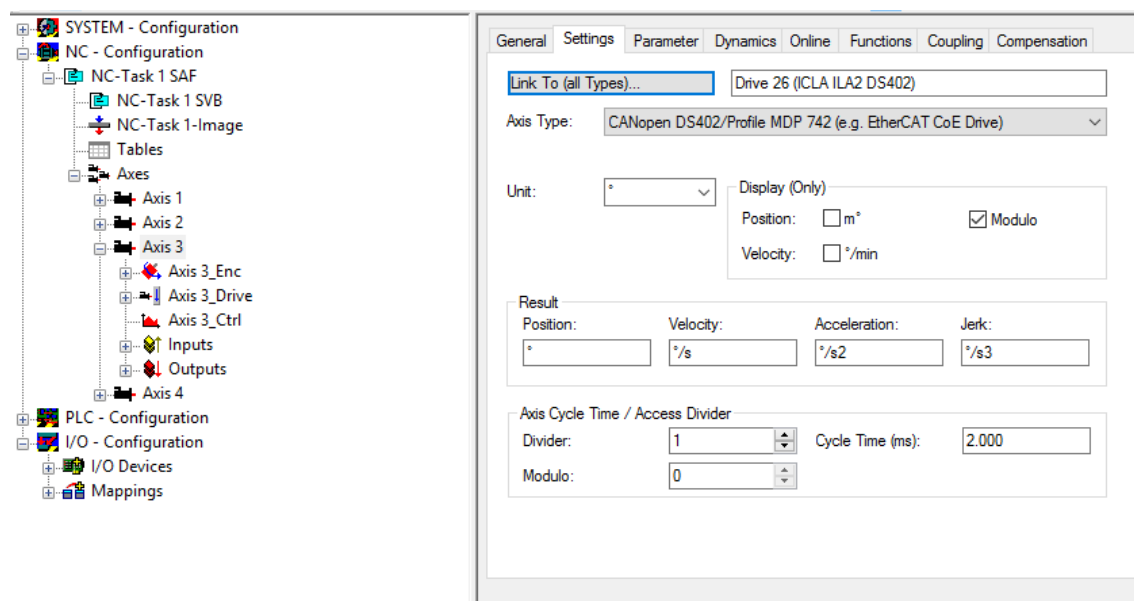
Další parametr, který musíme pro tyto dvě osy změnit je Scaling factor. Jedná se o měřítko, které reprezentuje počet kroků (impulzů) na jednu otáčku motoru. To může mít každý typ motoru jiné. U našeho motoru se position scaling rovná 32768 krokům na jednu otáčku. Jelikož jedna otáčka odpovídá 360° tak pomocí následujícího vztahu (5.1) vypočítáme náš potřebný Scaling factor.

$$Scaling\ factor = \frac{1}{\frac{Počet\ kroků\ na\ otáčku}{Otáčka\ [°]}} = \frac{1}{\frac{32768}{360°}} = 0,010986328125\ °/krok$$

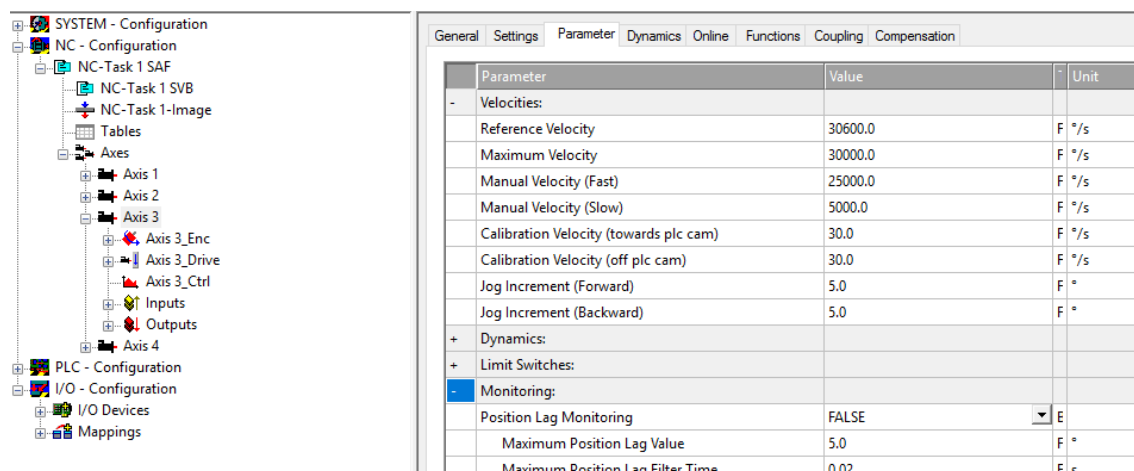
(5.1)



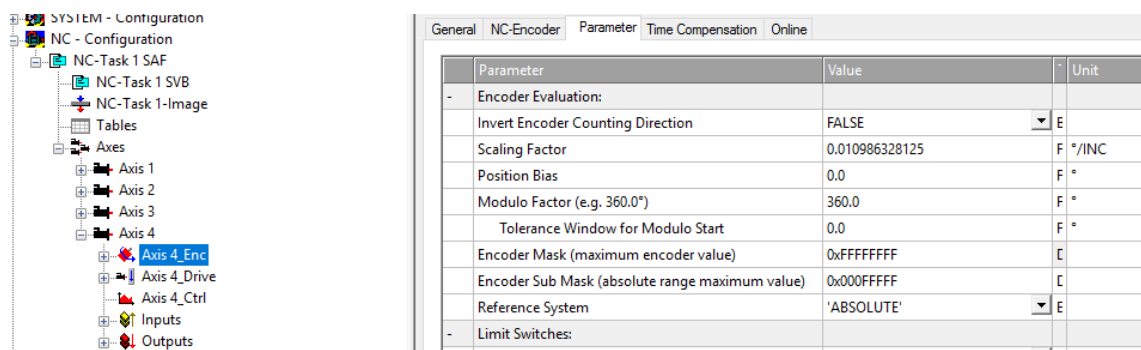
Tento parametr je potřeba zadat do parametrů pro enkodér. K těm se dostaneme opět přes „Axis“, kterou rozklikneme a uvidíme „Axis\_Enc“. Na tu klikneme a v pravé části se nám opět zobrazí okno. Vybereme záložku „Parameter“ a v kolonce „Encoder Evaluation“ nastavíme námi vypočítanou hodnotu do parametru „Scaling Factor“ (Obr. 38).



Obr. 36 - Nastavení jednotek a „Display(Only)“



Obr. 37 - Nastavení otáček a monitorování



**Obr. 38 - Nastavení Scaling Factor**

Axis (1,2) - U těchto os v podstatě není nutné měnit žádné parametry. Jelikož se jedná o osy, které provádí lineární pohyb, můžeme nechat všechny parametry tak, jak jsou nastavené v základu. Pro upřesnění tyto osy mají nastavené jednotky mm, v části Display(Only) se nic nezaškrťává, Position lag monitor může zůstat zapnutý, jelikož u těchto os nebudeme dosahovat rychlostí takových, aby překročily nastavený maximální position lag. Referenční a maximální otáčky také můžeme nechat tak, jak jsou nastaveny opět z důvodu nedosahování vysokých rychlostí u těchto os. Jejich Scaling factor je nastaven na 0,0001 mm/INC.

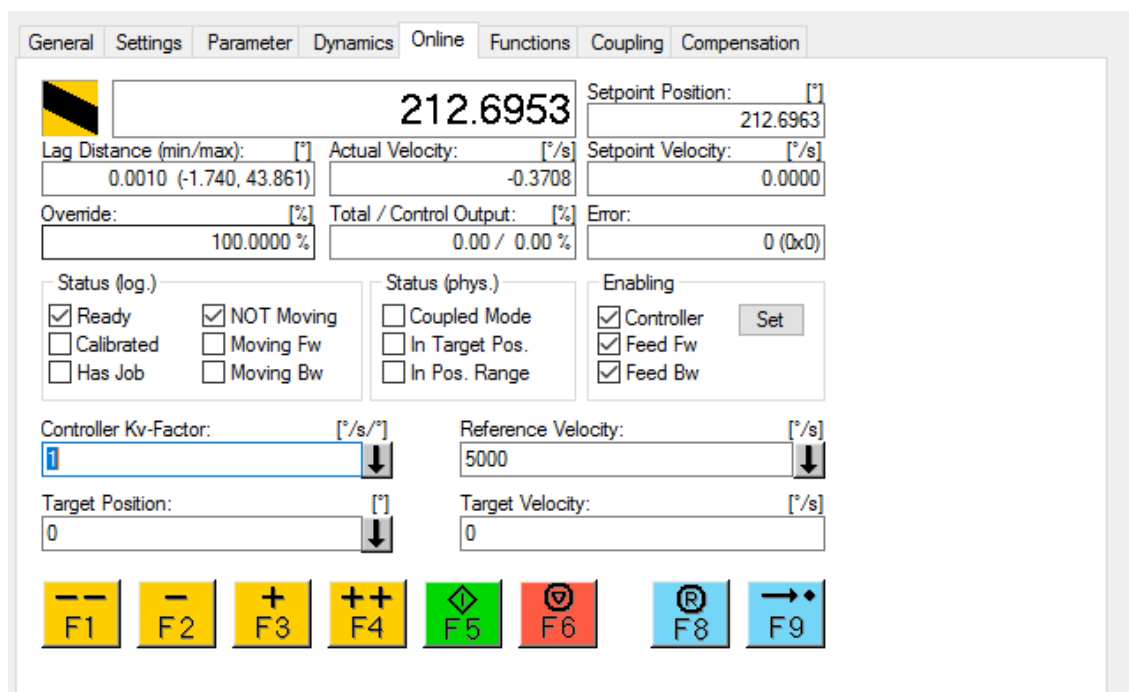
Nyní již máme vše připravené a můžeme se pustit buď do manuálního ovládání, které prostředí System Manager dovoluje, nebo si naprogramovat vlastní program v programu TwinCAT PLC Control.

## 5.2 Manuální ovládání os

Po předchozím nastavení všech potřebných parametrů nabízí System Manager ovládání jakékoliv osy manuálně bez vytváření jakýchkoliv programů.

Pro použití manuálního ovládání zvolíme položku „NC - Configuration“, kterou rozklikneme a dále rozklikneme NC úlohu („NC – Task 1 SAF“). Pak rozklikneme „Axes“ a vybereme si osu, se kterou chceme manuálně pohybovat. Klikneme tedy na vybranou osu a v pravé části prostředí se nám nabídne několik záložek. Zvolíme záložku „Online“, kde se manuální ovládání nachází.

Vybranou osu je dále v záložce „Online“ potřeba povolit. To se provede kliknutím na tlačítko „Set“ v části „Enabling“. Vyskočí okno, kde klikneme na tlačítko „All“, které nám zapne plnou kontrolu a potvrdíme tlačítkem „OK“.



Obr. 39 - Manuální ovládání os v TwinCATu

Na Obr. 39 lze vidět prostředí pro manuální ovládání. Tlačítka F1 a F4 slouží pro rychlý manuální pohyb, F2 a F3 pro pomalý manuální pohyb osou. Tlačítkem F8 lze resetovat osu v případě, že nastane nějaká chyba (Error).

Další možností online manuálního pohybování je možnost používat jednotlivé funkce, které nám potom nabízí samotné programování. Tato možnost se nachází hned vedle záložky „Online“ a je to záložka „Functions“. Je zde tak možnost natočit osu do nějaké absolutní polohy, nebo jen relativně apod.

Další možností vyzkoušení si funkce, kterou budeme potřebovat pro náš model, je možnost spojování (coupling) motorů. Tato funkce se nachází pod záložkou „Coupling“. Zde je možnost vybrat si osu, která bude pro právě zvolenou osu jako Master. Z naší osy tak v podstatě uděláme Slave osu, která se bude řídit podle Master osy. Chování Slave osy se pak dá ovlivnit pomocí parametru „Coupling factor“, který určuje, jakým způsobem bude Slave osa Master osu kopírovat. Např. hodnotou -1 se bude Slave osa točit stejnou rychlostí, avšak opačným směrem. To využijeme při programování právě pro osy s ozubenými koly, kde potřebujeme, aby se motory točily stejnou rychlostí ale opačným směrem.

## 5.3 PLC Control

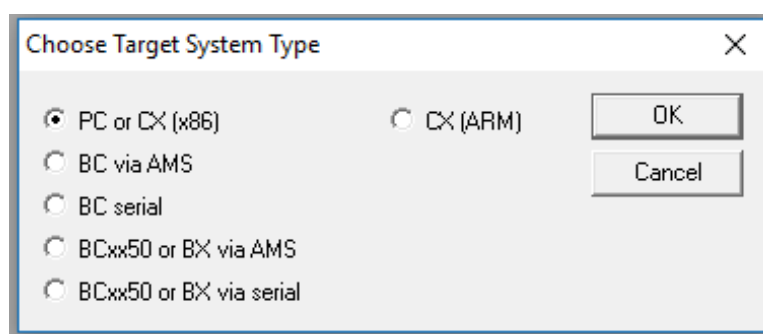
Tento modul je vývojové prostředí, které je určené pro programování řídicích programů pro PLC a to dle standardu IEC 61131. V podstatě se jedná o program CoDeSys, který je ukrytý právě v prostředí TwinCAT.

PLC Control obsahuje všechny programovací jazyky definované standardem IEC 61131-3.

- IL – Instruction list
- FBD – Function block diagram
- LD – Ladder diagram
- SFC – Sequential function chart
- ST – Structured text
- CFC – Continuous function chart

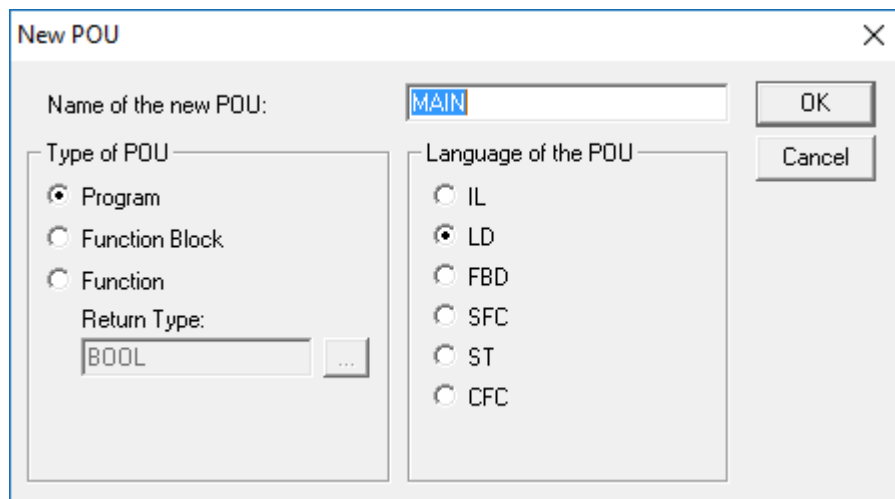
### 5.3.1 Spuštění PLC Control

Při prvním spuštění tohoto programovacího prostředí se nám začne vytvářet nový projekt. Jako první je třeba si vybrat, jaký systém používáme, v našem případě se jedná o PC, zvolíme tedy první možnost (Obr. 40).



Obr. 40 - Výběr typu systému

Dalším krokem je nové „POU“ (Programová organizační jednotka), kde typ POU vybereme „Program“. Ten vrátí jednu nebo několik hodnot během provozu a všechny hodnoty jsou zachovány z poslední doby, kdy byl program spuštěn. Nazveme si ho jako „MAIN“, bude se tedy jednat o hlavní program a zvolíme si programovací jazyk (Obr. 41).

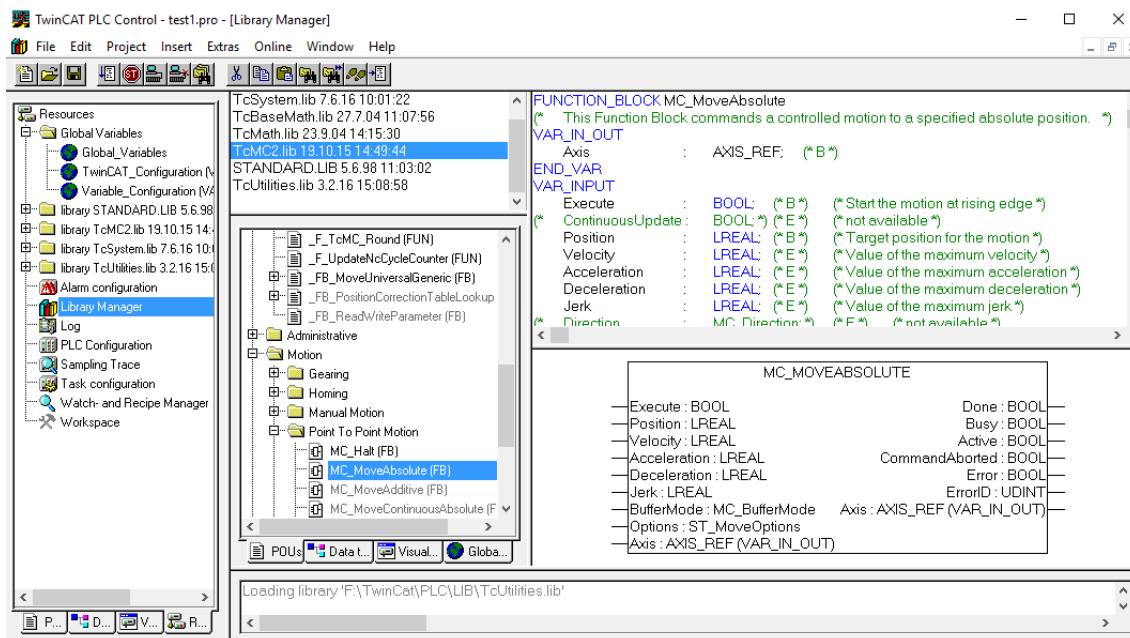


Obr. 41 - Vytvoření nového POU

Když potom klikneme v záložce „POUs“ pravým tlačítkem, můžeme si přidávat další programy, funkce nebo funkční bloky.

### 5.3.2 Příprava knihoven

V levé části prostředí se nachází okno, ve kterém jsou záložky jako „POUs“, „Data type“, „Visualizations“ a „Resources“. V záložce „Resources“ se potom nachází „Library manager“, který dvojklikem rozklikneme a otevřou se nám v pravé části prostředí okna ke knihovnám. V horní části (Obr. 42) jsou umístěny jednotlivé knihovny, které už jsou automaticky přidány. Další knihovnou, kterou budeme potřebovat je knihovna „TcMC2.lib“. Tato knihovna je už připravena v místě instalace programu, jen je potřeba si jí přidat. Klikneme tedy do tohoto okna pravým tlačítkem a zvolíme „Additional library“. Otevře se nám okno, ve kterém je nutno najít cestu do umístění knihoven v místě nainstalování programu. Jakmile se dostaneme do daného adresáře tak si najdeme potřebnou knihovnu a klikneme na „Otevřít“. Tím se nám knihovna přidá do seznamu. Společně s ní se přidá ještě několik dalších knihoven, které jsou nejspíše s touto knihovnou spřažené.



Obr. 42 - Přidání a prohlížení knihoven

Když nyní na přidanou knihovnu klikneme, v části pod seznamem knihoven se nám zobrazí vše, co tato knihovna nabízí. Můžeme si tak vše projít a podívat se k čemu jaký blok v dané knihovně slouží a jaké má parametry.

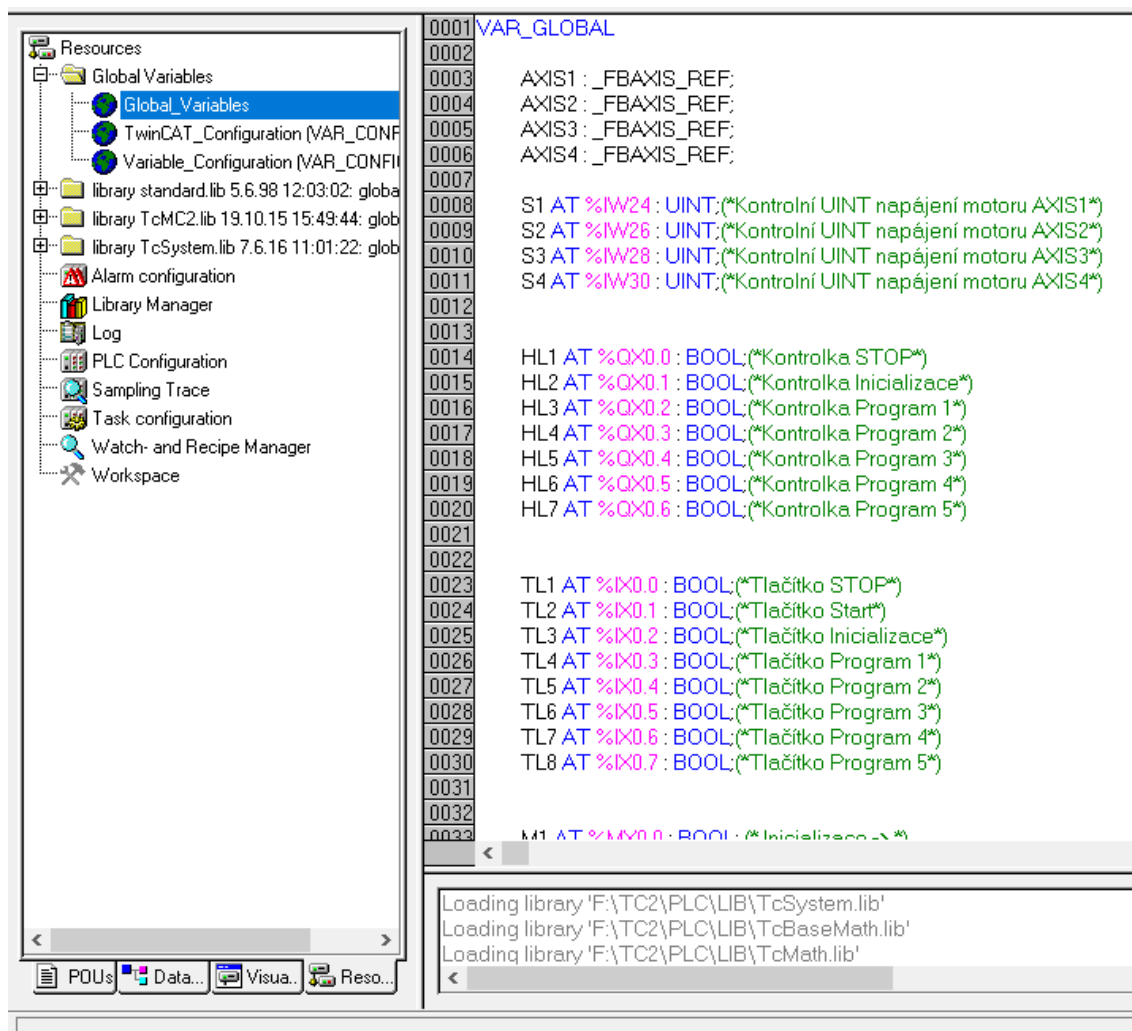
Pokud by byla potřeba nějaká další knihovna tak stejným postupem přidáme další knihovnu. V případě, že není dostupná v základní složce s knihovnami nainstalovanými při instalaci TwinCATu, je nutné si knihovnu nejprve stáhnout popř. koupit.

Nyní už v podstatě můžeme přejít na samotné programování.

### 5.3.3 Programování

Programování se provádí v záložce „POUs“, kde máme v základu pouze „MAIN“, který jsme si vytvořili při zakládání nového projektu, a slouží jako hlavní program.

V záložce „Resources“ je dobré si v kolonce „Global Variables“ nadefinovat všechny osy (Obr. 43). Dále si zde můžeme nadefinovat vstupní a výstupní adresy pro kontrolky a tlačítka a vstupní parametry UINT pro kontrolu napájení jednotlivých motorů. V další části těchto proměnných jsou potom již připraveny merkery pro programování.



**Obr. 43 - Globální proměnné**

Vstupní a výstupní proměnné a nadefinované osy se nám potom zobrazí v prostředí System Manager, kde k nim přiřadíme jednotlivé osy „Axis“, I/O moduly a další parametry motorů.

### 5.3.3.1 Knihovna TcMC2

Pomocí této knihovny je realizováno řízení os. Popíšeme si jednotlivé bloky, které tato knihovna obsahuje a my je použijeme v programování.

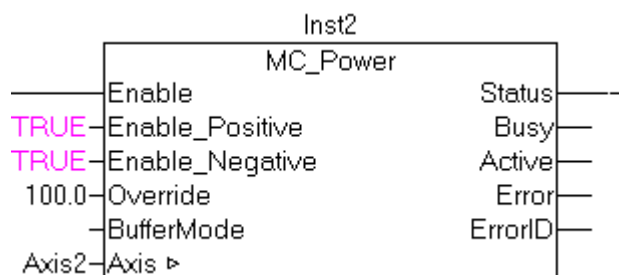
#### MC Power

Jedná se v podstatě o nejdůležitější blok, bez kterého by to nefungovalo. Tento blok (Obr. 44) v podstatě zajišťuje povolení osy, obdobně jako povolení v případě manuálního ovládání. Má jak vstupní tak i výstupní proměnné.

Jako vstupní má Enable (software povolení osy), Enable\_Positive, Enable\_Negative (povolení osy v daných směrech), Override (nastaveno na 100%).

Jako výstupní jsou potom Status (informuje o tom, zda je nebo není daný blok aktivní) Busy, Active a Error jsou také boolovské proměnné, které informují o stavu a ErrorID je typu UDINT a vrací nám tedy označení erroru.

Axis je potom vstupně-výstupní proměnná a zde se vloží osa, kterou jsme si nadefinovali do globálních proměnných a chceme jí tímto blokem povolit.

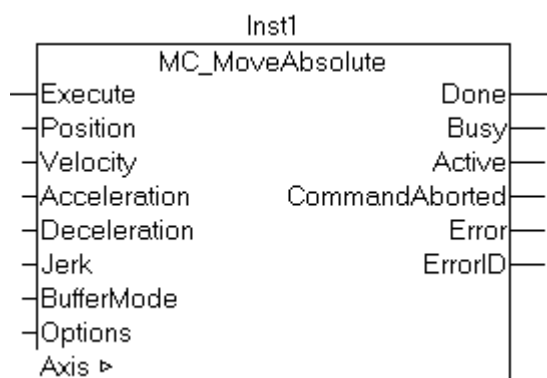


**Obr. 44 - Funkční blok MC\_Power**

### MC MoveAbsolute

Tento blok (Obr. 45) slouží pro pohybování osy pomocí absolutního polohování. Jelikož servomotory vědí, kde mají nulovou polohu, je tak možné pomocí tohoto bloku nastavit motor do požadované pozice.

Execute slouží pro vykonání tohoto bloku. Jakmile bude na tomto vstupu log. 1, začne se provádět úkon, který má tento blok udělat a to dostat se do nastavené polohy (Position). Jakmile se tedy dostane do zadané polohy, výstup Done změní svůj stav z log. 0 na log. 1. Absolutní polohování lze ještě vysvětlit tak, že když se dostaneme do zadané polohy a spustíme tento blok znovu se stejnou polohou, nestane se nic, protože se již v této poloze nachází.



**Obr. 45 - Funkční blok MC\_MoveAbsolute**

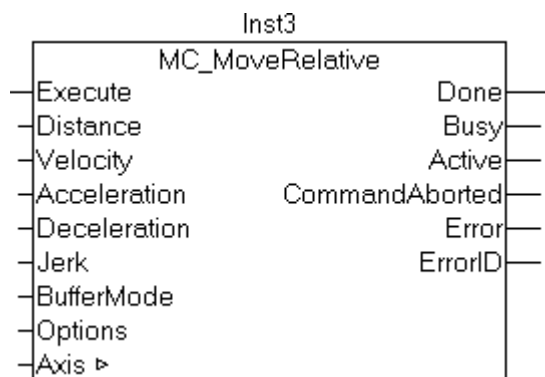
### MC MoveRelative

Tento blok (Obr. 46) slouží opět pro pohybování osy, avšak již pomocí relativního polohování. Tedy tento blok pootočí motorem o nastavenou vzdálenost (Distance). Bude-li se motor nacházet v poloze 20 a nastavená vzdálenost tohoto bloku bude 20,



motor se při spuštění této funkce pootočí o 20 a výsledná poloha bude 40. Při dalším spuštění bude výsledná poloha 60 apod.

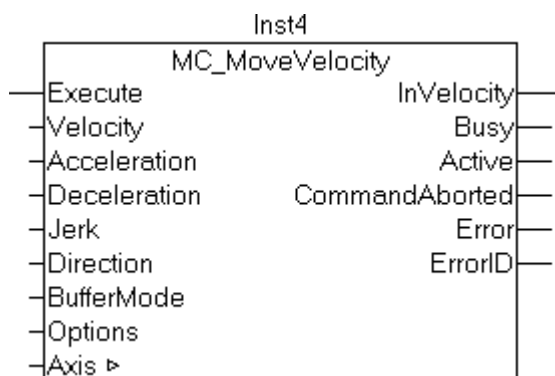
Tento blok se opět spustí přivedením log. 1 na vstup Execute a po dokončení činnosti výstup Done opět změní svůj stav na log. 1.



**Obr. 46 - Funkční blok MC\_MoveRelative**

### MC MoveVelocity

Tento blok (Obr. 47) slouží pro roztočení motoru na danou rychlost. Zde je tak důležitým vstupním parametrem vstup Velocity. Ten určuje otáčky, kterými chceme, aby se motor točil. Vstup Execute nám opět spustí změnou na log. 1 tento blok. Jakmile motor dosáhne nastavených otáček, výstup InVelocity změní stav na log. 1.



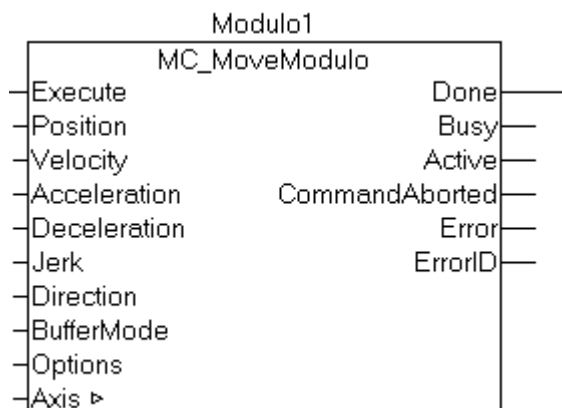
**Obr. 47 - Funkční blok MC\_MoveVelocity**

### MC MoveModulo

Tento blok (Obr. 48) je určen pro rotaci motoru do jisté modulo pozice. Využijeme ho hlavně při inicializaci, kdy požadujeme určitý stupeň natočení motorů. Tento blok pootočí celý motor maximálně o jednu otáčku do požadované pozice. Což je hlavním důvodem jeho použití. Blok MC\_MoveAbsolute totiž otáčí motorem tak dlouho, dokud nedosáhne absolutní nuly.

Parametr Position udává, na jakou hodnotu chceme motor natočit, Velocity jsou otáčky, kterými se může motor točit, Acceleration a Deceleration pak pro zrychlení a zastavení motoru. Do Axis napíšeme jednu z našich os.

Příkaz se spustí při log. 1 na vstupu Execute. Jakmile blok dokončí svojí funkci, výstup Done bude mít hodnotu log. 1.



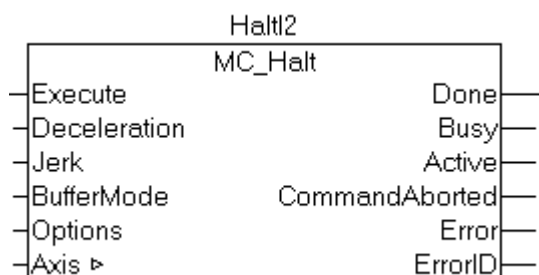
**Obr. 48 - Funkční blok MC\_MoveModulo**

### MC\_Halt

Tento blok (Obr. 49) je určen pro zastavení motoru s možností dalšího roztočení motoru. Jelikož jakmile se spustí jakýkoliv z bloků MC\_MoveAbsolute, MC\_MoveRelative, MC\_MoveVelocity, nelze je zastavit změnou vstupního parametru Execute na log. 0.

U bloků MC\_MoveAbsolute a MC\_MoveRelative to většinou moc nevádí, jelikož nastavenou hodnotu polohy popř. vzdálenosti máme jistě odzkoušenou a chceme, aby motory těchto pozic dosáhly. Jakmile jich dosáhnou tak se zastaví.

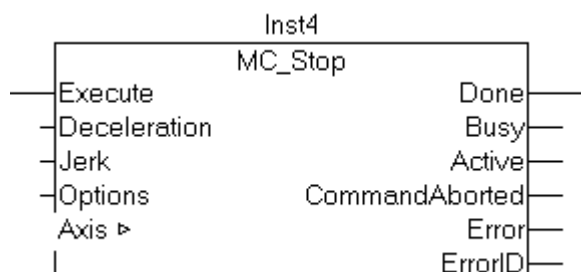
Blok MC\_MoveVelocity se však dosažením daných otáček nezastaví. To by ani nebyl chtěný prvek. Pro jeho zastavení lze pak použít právě MC\_Halt. Hodnotou Deceleration lze pak nastavit, jak rychle se zastaví.



**Obr. 49 - Funkční blok MC\_Halt**

### MC Stop

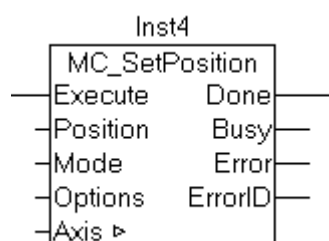
Tento blok (Obr. 50) slouží k okamžitému zastavení motoru. Slouží primárně pro nouzové zastavení motoru. Zastavení se provede v případě, že na vstup Execute přijde log. 1. Parametrem Declaration si potom můžeme nastavit, jak rychle by měl motor zastavit. Parametr Axis je potom osa, kterou chceme zastavit.



**Obr. 50 - Funkční blok MC\_Stop**

### MC SetPosition

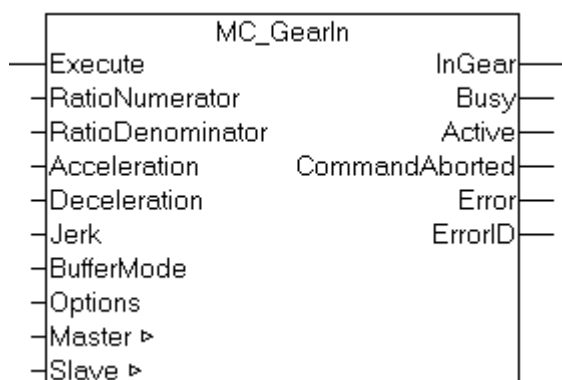
Tímto blokem (Obr. 51) lze změnit motoru jeho absolutní polohu. Ne však tím, že by se do ní motor dotočil, ale tím, že se tato hodnota pouze přepíše. Vstupní parametr Position si nastavím třeba na 0 a jakmile se Execute změní na log. 1, přepíše se mi pozice dané osy. Toto lze využít např. pro nastavení referenční polohy motorů.



**Obr. 51 - Funkční blok MC\_SetPosition**

### MC GearIn

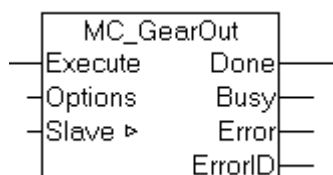
Tento blok (Obr. 52) slouží pro spojení neboli lineární coupling slave osy k ose master. Spojení lze provést pouze v případě, že obě dvě osy neprovádí žádný pohyb. Jakmile dojde ke spojení, tak lze dávat povely pouze master ose. Když přijde povel pro osu slave, tak ten příkaz osa neprovede. Vstupy „Ratio“ slouží pro určení poměru, v jakém bude osa slave kopírovat osu master. „Numerator“ je číselník a „Denominator“ jmenovatel. Opět dojde k provedení této funkce, jakmile se na vstupu „Execute“ objeví log. 1. Jakmile dojde ke spojení os, výstup „InGear“ změní svůj stav na log. 1.



**Obr. 52 - Funkční blok MC\_GearIn**

### MC GearOut

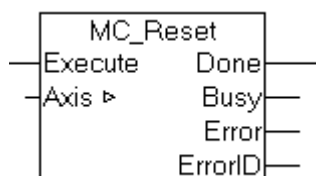
Tento blok (Obr. 53) je potom určen pro zrušení spojení (Couplingu) mezi master a slave osou. Tedy slave osa se odpojí od master osy a opět můžeme manipulovat s každou zvlášť. Zde je už jako parametr pouze slave osa. A to z důvodu, že jen ta byla napojena na master osu. Je nutné tak pouze odpojit slave od master nikoliv naopak. K odpojení dojde, jakmile se na vstupu „Execute“ změní logický stav z 0 na 1. Že došlo ke správnému odpojení a odpojení proběhlo, nám oznámí výstupní parametr „Done“ jehož hodnota bude log. 1.



**Obr. 53 - Funkční blok MC\_GearOut**

### MC Reset

Tento funkční blok (Obr. 54) slouží pro resetování errorů os. V případě, že tedy nějaká osa spadne do stavu, který ohlásí error, zavoláním tohoto bloku by se měla osa resetovat zpět do fungující osy. Změnou vstupní proměnné „Execute“ na log. 1 tedy dojde k resetování. Úspěšné resetování potom ohlásí výstupní proměnná „Done“.



**Obr. 54 - Funkční blok MC\_Reset**

### 5.3.3.2 Vlastní funkce

Vytvořil jsem si dva jednoduché vlastní funkční bloky „Porovnani“ a „Nerovnost“ (Obr. 55, Obr. 56). Funkční blok „Porovnani“ porovnává dvě vstupní hodnoty typu UINT a výstupem je potom hodnota typu BOOL.

0001	FUNCTION_BLOCK Porovnani
0002	VAR_INPUT
0003	X: BOOL;
0004	A: UINT;
0005	B: UINT;
0006	END_VAR
0007	VAR_OUTPUT
0008	Y: BOOL;
0009	END_VAR
0010	VAR
0011	END_VAR
0012	<
0001	IF (X = TRUE AND A = B)
0002	THEN Y := TRUE;
0003	ELSE Y := FALSE;
0004	END_IF
0005	
0006	

Obr. 55 - Funkční blok Porovnani

Tuto funkci potom používám pro hlídání toho, zda jsem s motory lineárního pohybu dojel na referenční snímač. Tato hodnota je z motoru vysílána v datovém typu UINT. Nelze získat přímo BOOL hodnotu změny stavu snímače. Název této proměnné v motoru je \_IO\_LIO\_act. Zjistil jsem si, jakou hodnotu má, je-li snímač aktivní a tu jsem použil jako jednu vstupní proměnnou tohoto funkčního bloku. Jako druhou pak používám aktuální hodnotu. Jakmile najede motor na referenční snímač, hodnoty jsou si rovny a na výstupu tohoto funkčního bloku dostanu log. 1.

Další její použití je potom v kontrole toho, jestli jsou vůbec motory zapojeny v napájení. To je parametr „State“, který ukazuje aktuální stav motoru.

Funkční blok „Nerovnost“ (Obr. 56) je potom v podstatě opakem funkčního bloku Porovnani. Tedy pokud je parametr A různý od parametru B, je výstup v log. 1. Toto lze použít při kontrole toho, zda nedošlo k odpojení motorů z napájení.

0002	VAR_INPUT
0003	X: BOOL;
0004	A: UINT;
0005	B: UINT;
0006	END_VAR
0007	VAR_OUTPUT
0008	Y: BOOL;
0009	END_VAR
0010	VAR
0011	END_VAR
0012	
	< [ ]
0001	IF (X = TRUE AND A <> B)
0002	THEN Y := TRUE;
0003	ELSE Y := FALSE;
0004	END_IF
0005	

Obr. 56 - Funkční blok Nerovnost

### 5.3.3.3 Vytváření proměnných

Všechny merkery, vstupní a výstupní proměnné jsem si vytvořil v globálních proměnných. Stejně tak i osy. Jednotlivé instance jsem si potom vytvořil vždy uvnitř daného funkčního bloku. Všechny globální proměnné jsou k nalezení v příloze 2.

Zápis pro merker:

M1 AT %MX0.0 : BOOL;

Zápis pro vstupní nebo výstupní BOOL hodnotu:

HL1 AT %QX0.0 : BOOL;

TL1 AT %IX0.0 : BOOL;

Zápis pro UINT hodnotu (použité pro referenční snímače)

SQ3 AT %IW4 : UINT;

Vytváření instancí pro funkční bloky

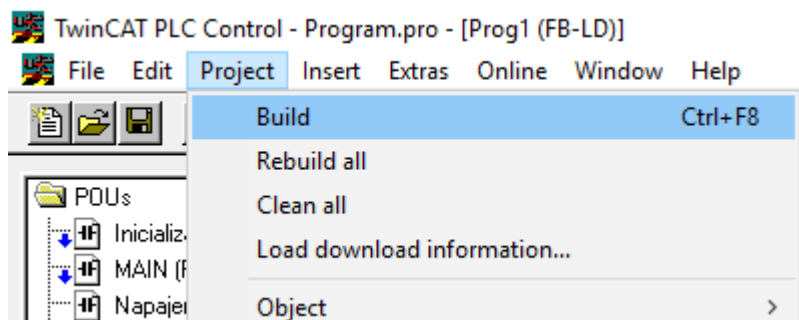
Název : Název funkčního bloku;

Např: Power1 : MC\_Power;

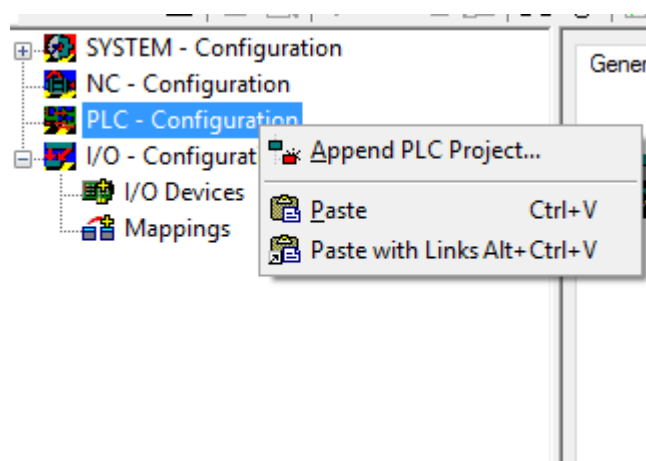
### 5.3.3.4 Připojení programu z PLC Control do System Manageru

Jakmile máme hotový projekt v PLC Control resp. napsaný program, musíme nejprve vytvořit soubor, který lze potom připojit v System Manageru. To se provede v PLC Control buď klávesovou zkratkou ctrl+F8, nebo kliknutím na záložku „Project“ v horní liště a kliknutím na „Build“ (Obr. 57). Tím se nám vytvoří soubor (.tpy), který potom můžeme připojit v System Manageru. To se provede tak, že si v System Manageru klikneme na položku „PLC - Configuration“ pravým tlačítkem a dáme „Append PLC Project...“ (Obr. 58). Dojde k otevření okna, kde si musíme najít cestu

tam, kam jsme si uložili PLC projekt z PLC Control. Po najití ho vybereme a klikneme „Otevřít“.



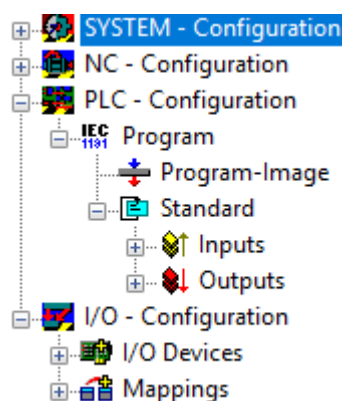
Obr. 57 - Vytvoření souboru pro připojení projektu v System Manageru



Obr. 58 - Připojení PLC programu k projektu v System Manageru

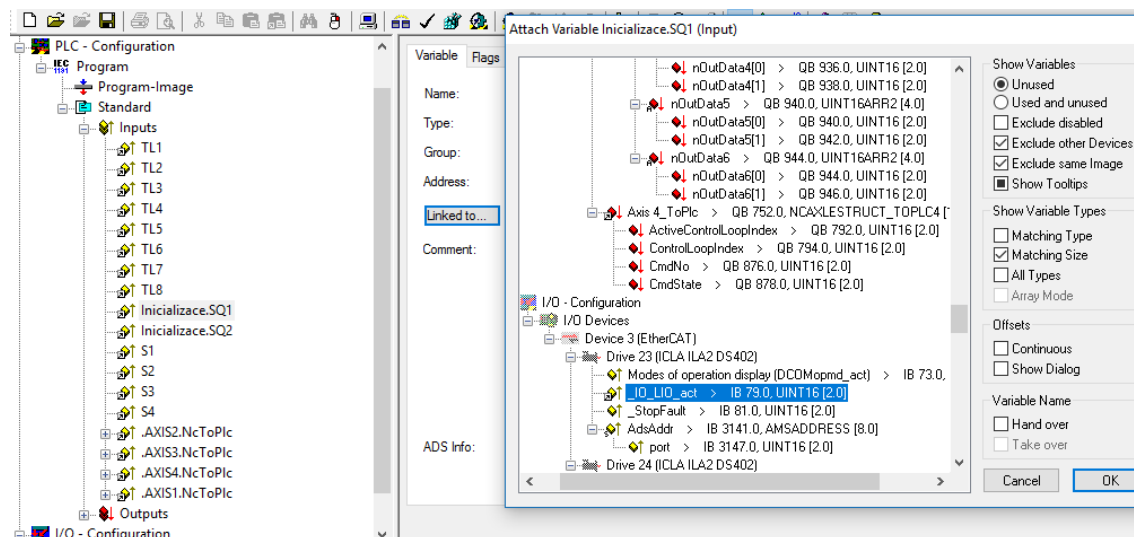
### 5.3.3.5 Linkování proměnných v System Manageru

Vytvořené proměnné, které je nutné spojit s reálnými prvky, se spojují v System Manageru. Jakmile máme připojený PLC projekt, lze ho rozkliknout a v položce „Standard“ jsou potom vstupy a výstupy (Obr. 59), které je možné propojit právě z PLC programu do System Manageru.



Obr. 59 - Vstupy a výstupy PLC programu

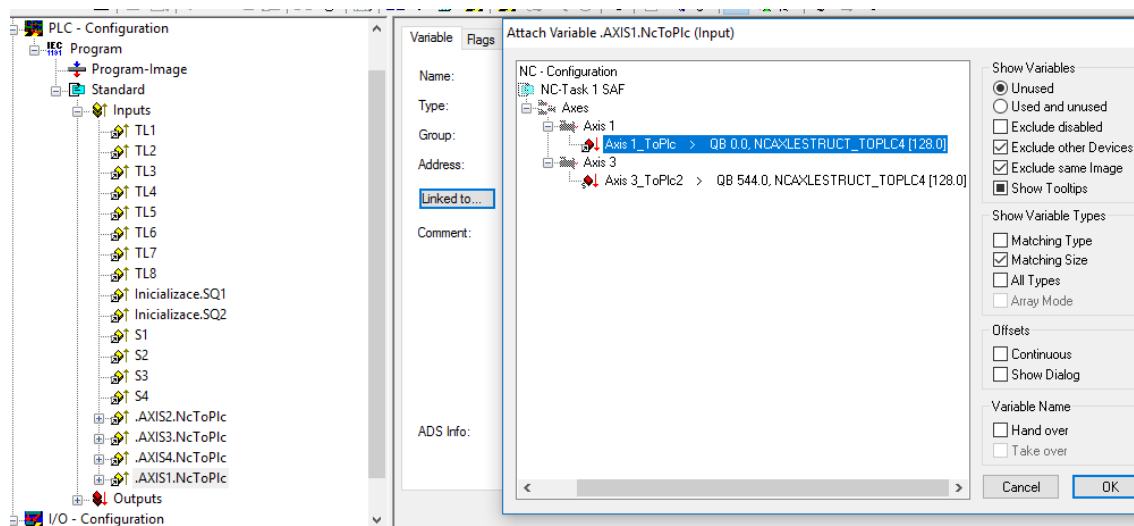
Příklad nalinkování proměnné SQ1, kterou chceme jako referenční snímač osy Axis 1, je zobrazen na Obr. 60. Je nutné si tedy ve vstupních proměnných rozkliknout danou proměnnou, v našem případě SQ1. Otevře se nám okno v pravé části prostředí, kde klikneme na „Linked to...“. Otevře se nám okno se všemi dostupnými proměnnými, na které lze naši proměnnou připojit. My však hledáme proměnnou `_IO_LIO_act`. Ta je k dispozici v `Driverech`, a pro naší Axis 1 je odpovídající Drive 23. V něm tedy nalezneme tuto proměnnou, vybereme jí a klikneme na „OK“.



**Obr. 60 - Linkování proměnné SQ3 na referenční snímač**

Nalinkování jednotlivých Axis probíhá podobně. Axis je nutné nalinkovat jak ve vstupních tak i ve výstupních proměnných. Vybereme si tedy například Axis2. Opět na ní klikneme a klikneme na „Linked to...“. Zde se potom musí vybrat osa jako celek, viz Obr. 61. U Axis se ale na vstupní proměnnou z PLC programu linkují výstupní z System Manageru a naopak. To je dáno směrem šíření informací. PLC bere informaci z Axis do PLC jako vstupní, zatímco System Manager bere informaci z Axis do PLC jako výstupní informaci z dané Axis.





Obr. 61 - Linkování Axis

### 5.3.4 PLC Control Online režim

Máme-li vše připravené, tedy nakonfigurovaný TwinCAT, napsaný program připojený do System Manageru a nalinkované proměnné, můžeme přejít k otestování našeho programu.

Nahrání a spuštění programu je možné v záložce „Online“, na kterou klikneme a nejprve klikneme na „Login“. Tím se připojíme k našemu IPC. Zeptá se nás, zda chceme náš program nahrát do IPC. To potvrdíme, jinak by to nefungovalo.

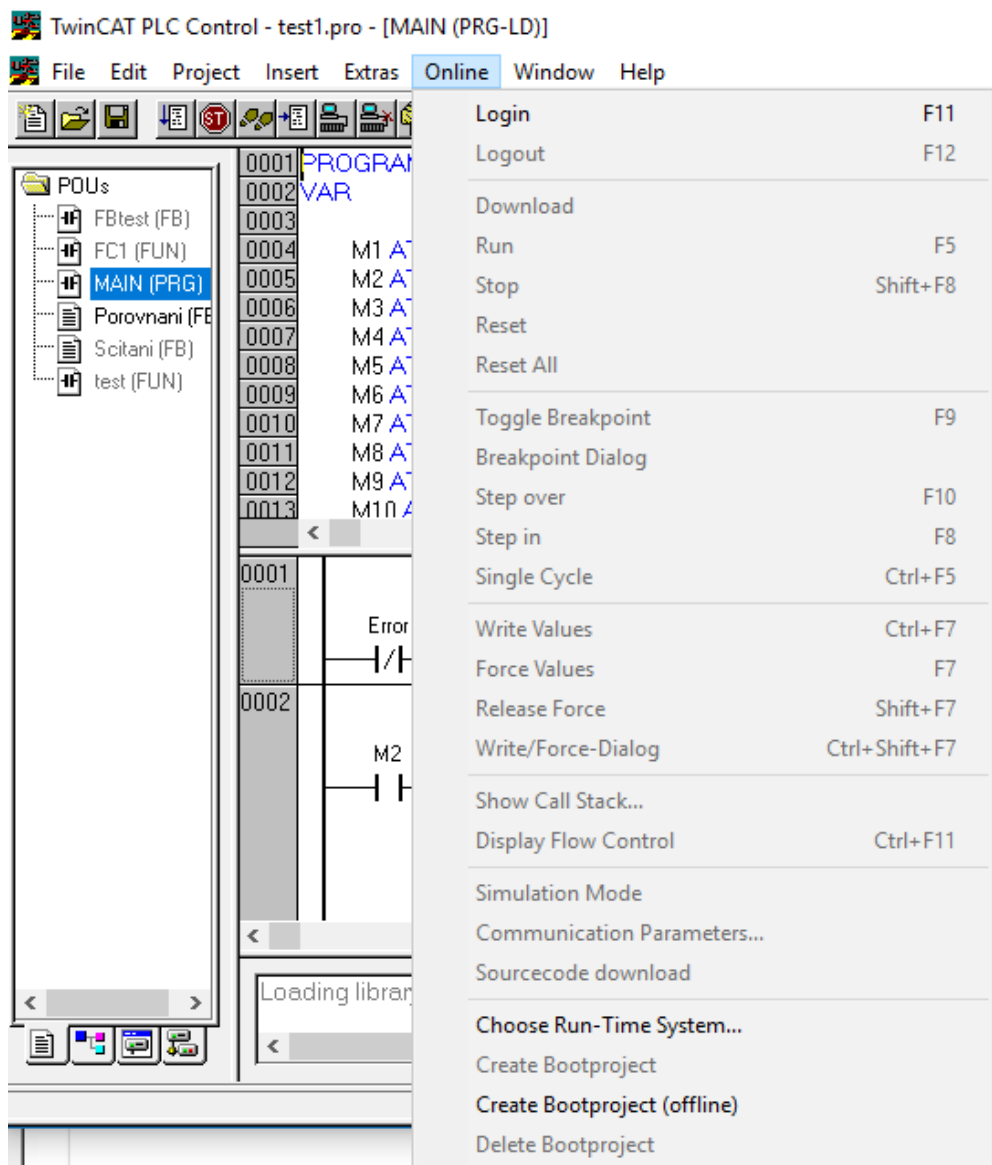
Nyní již můžeme přejít k testování resp. ke spuštění našeho programu. Přepneme se v záložce „Online“ do „Run“ módu. Jakmile toto uděláme, v okně s proměnnými lze vidět jejich aktuální stav. Pokud program ovládáme pomocí merkerů, dvojklikem na daný merker se nám v okně s proměnnými ukáže, do jakého stavu přejde po potvrzení. Je zde možnost buď „True“ nebo „False“. Navolíme si tedy požadovanou hodnotu a klávesovou zkratkou ctrl+F7 se přepíše jeho hodnota. Takto je možné tedy testovat program. Pokud máme ovládání připraveno pro tlačítka, tak pro spuštění používáme je.

Pokud chceme uložit celý program do IPC natrvalo, tedy že se uloží do trvalé paměti a ne jen do krátkodobé, která se po odpojení napájení vymaže, je potřeba opět v záložce „Online“ být jednak přihlášen k IPC a v případě, že jsme přihlášení, klikneme na „Create Bootproject“. Tato možnost nám rovnou i umožní nahrání do trvalé paměti zařízení. Pokud potom chceme, aby se nám daný program i celá konfigurace spouštěla sama po zapnutí celého zařízení do napájení, je potřeba v prostředí System Manager nejprve aktivovat aktuální konfiguraci a potom v položce „SYSTEM - Configuration“ v pravé části prostředí v záložce „Boot Settings (Target)“ zvolit v části „Boot Settings“ „Run Mode (Enable)“ a to potvrdit kliknutím na tlačítko „Apply“ (Obr. 63). Toto nastavení je chráněno přihlašovacím jménem a

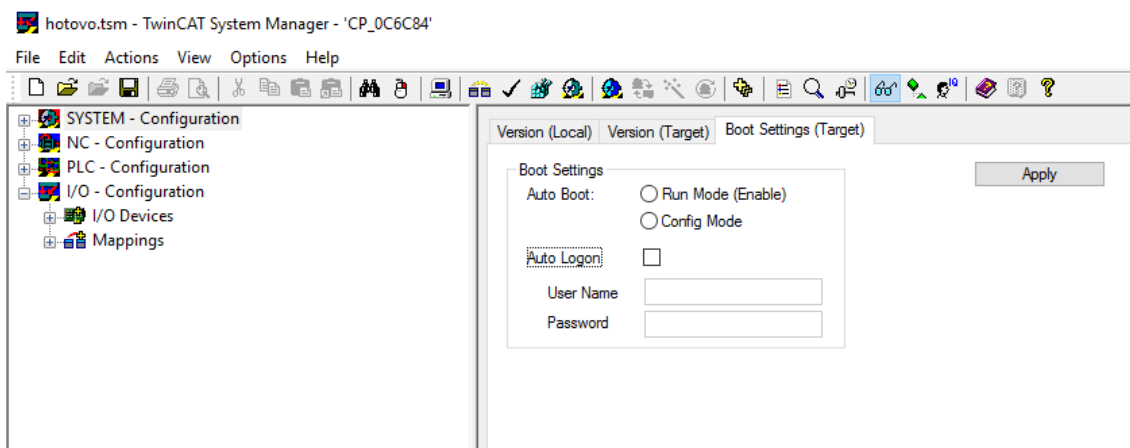
heslem, které bude zařízení požadovat hned po kliknutí na „Apply“. Vyskočí přihlašovací okno a přihlašovací údaje jsou:

- User name – admin
- Password - admin

Po skončení s online režimem se od IPC odhlásíme a to tak, že v záložce „Online“ klikneme na „Logout“. Všechny možnosti, které záložka „Online“ nabízí lze vidět na Obr. 62.



Obr. 62 - Záložka online a její možnosti (PLC Control)



Obr. 63 - Boot Settings

### 5.3.5 Vlastní program

Vytvořil jsem program, který je určen pro ukázkou synchronizace motorů. Program obsahuje 5 různých scénářů pohybů motorů. Jednotlivé scénáře jsou spouštěny pomocí tlačítek z ovládacího panelu. Pro přehlednost je každý program umístěn ve svém funkčním bloku. Dalším funkčním blokem je potom i napájení motorů a inicializace modelu. Všechny funkční bloky se potom spouštějí za různých podmínek v hlavním programu MAIN.

Celý program (soubory pro PLC Control i pro System Manager) je umístěn v příloze 11. Jelikož popis jednotlivých sekvencí pohybů označených Prog1-Prog5 má vždy stejný základ, tak v části popisující program popisují pro příklad pouze Prog1 a Prog4. Výpisy jednotlivých programů jsou umístěny v přílohách 3 – 10.

#### 5.3.5.1 Popis funkce jednotlivých předváděcích programů

##### Prog1

Dojde ke spojení os s ozubenými koly. Následně se tyto osy roztočí na dané otáčky a dojde ke splynutí ozubených kol pomocí pohybu lineárních os. Po 2 sekundách od sebe zase odjedou.

##### Prog2

Dojde ke spojení os s ozubenými koly. Opět se roztočí na dané otáčky a budou se k sobě přibližovat rozdílnými rychlostmi. Jenda lineární osa pojedou 3x rychleji než druhá. Ke splynutí ozubených kol tak dojde v jedné čtvrtině cesty pomalejší osy. Poté dojde k navýšení otáček ozubených kol. Lineární osy nyní pojedou ve stejném směru, avšak jedna osa pojedou pomaleji. Dojde tak při cestě k rozjetí ozubených kol

a na konci pohybu opět ke splynutí. Nakonec osa vzdálenější od svého referenčního snímače odjede k jeho blízkosti. Poté dojde k zastavení ozubených kol.

### **Prog3**

Dojde ke spojení os s ozubenými koly. Ozubená kola se roztočí na dané otáčky. Dojde ke splynutí ozubených kol pomocí lineárních os. Poté dojde k navýšení otáček ozubených kol a ke spojení lineárních os. Dojde k pohybu splynutých ozubených kol na jednu stranu modelu. Poté dojde k dalšímu navýšení otáček ozubených kol a pojedou splynuté na druhou stranu. Nakonec dojde k rozpojení lineárních os a odjetí vzdálenější osy k blízkosti svého referenčního snímače. Nakonec se zastaví i ozubená kola.

### **Prog4**

Dojde ke spojení os s ozubenými koly. Pomocí lineárního pohybu se k sobě přiblíží a dojde ke splynutí stojících ozubených kol. Na to dojde k roztočení ozubených kol. Ozubená kola od sebe odjedou a opět se zastaví. Následně se roztočí a opět splynou. Potom dojde k zastavení ozubených kol splynutých v sobě. Pojedou v sobě zastavené na jednu stranu a v průběhu pohybu se v sobě opět roztočí. Jakmile se lineární pohyb zastaví, dojde k zastavení ozubených kol. Nakonec od sebe ozubená kola odjedou.

### **Prog5**

Ozubená kola se rozjedou k sobě a v průběhu pohybu se roztočí. Dojde k jejich splynutí a pojedou spolu na jednu stranu. Budou spolu jezdit na jednu a na druhou stranu, zároveň se budou snižovat otáčky ozubených kol a zvyšovat rychlost lineárního pohybu.

#### **5.3.5.2 MAIN (Příloha 3)**

##### **Řádek 1**

Merker MN16 slouží jako informace o prvním stisknutí tlačítka START po zapojení modelu do napájení. Rozpínací kontakt MN8 udává informaci o tom, zda byl model zastaven tlačítkem STOP. Následující 4 bloky „Porovnání“ slouží pro kontrolu toho, že jsou motory v provozu a správném režimu. Tomu odpovídá právě hodnota 8 uvedená na vstupech B. Následujícím funkčním blokem je „Napájení“. Ten se spustí v případě, že všechny předešlé podmínky jsou splněny. Pokud dojde ke správnému zapnutí napájení motorů, dojde k setnutí stavu Nap\_OK.

## **Řádek 2**

Bloky „Nerovnost“ porovnávají, zda jsou motory ve funkčním stavu, čemuž odpovídá hodnota 8 na vstupu B. Je-li tomu tedy tak, neprojde signál přes tyto bloky. Pokud se však nějaký z motorů dostane z provozního režimu, dojde k resetu „Init\_OK.“

## **Řádek 3**

Stiskem tlačítka TL1 (STOP) dojde k setnutí merkeru MN10.

## **Řádek 4**

Merker MN11 – MN15 informují o stavu průběhu zastavení všech motorů po stisku tlačítka STOP. Jakmile jsou všechny aktivní, dojde k resetování hodnoty Nap\_OK a k setnutí merkeru MN8, jenž informuje o dokončení zastavení celého modelu.

## **Řádek 5**

Merker MN10 informující o stisku tlačítka STOP resetuje informaci o napájení Nap\_OK.

## **Řádek 6**

Po stisku tlačítka STOP dojde k zastavení osy AXIS1 pomocí bloku MC\_Stop. Po dokončení se setne MN11.

## **Řádek 7**

Po stisku tlačítka STOP dojde k zastavení osy AXIS2 pomocí bloku MC\_Stop. Po dokončení se setne MN12.

## **Řádek 8**

Po stisku tlačítka STOP dojde k zastavení osy AXIS3 pomocí bloku MC\_Halt. To z důvodu, že tato osa může být v průběhu stisku tlačítka STOP synchronizována s AXIS4. MC\_Halt zastaví v tomto případě obě osy, pokud jsou synchronizovány. Po dokončení se setne MN13.

## **Řádek 9**

Po stisku tlačítka STOP dojde k zastavení osy AXIS4 pomocí bloku MC\_Halt. V případě že jsou osy AXIS3 a AXIS4 synchronizovány, tak by zastavením osy AXIS4 pomocí MC\_Stop došlo k zastavení jen této osy, zatímco MC\_Halt nebude mít na osu AXIS4 vliv pokud je synchronizována s AXIS3. Po dokončení se setne MN14.

### **Řádek 10**

Po stisku tlačítka STOP a po dokončení zastavení osy AXIS3 dojde k ukončení synchronizace jak AXIS1 a AXIS2 tak i k ukončení synchronizace AXIS3 a AXIS4. AXIS 2 a AXIS 4 jsou vždy vedeny jako slave. Po dokončení se setne MN15.

### **Řádek 11**

Merker MN15 resetuje MN16.

### **Řádek 12**

Merker MN10 tedy stisk tlačítka STOP resetuje merkery MN1 – MN7, dále potom setne kontrolku HL1 odpovídající stavu STOP a resetuje všechny ostatní kontrolky. Ještě dojde k zresetování informace o inicializaci Init\_OK.

### **Řádek 13**

Tlačítko TL3 odpovídající inicializaci resetuje jednak informaci o inicializaci Init\_OK, dále potom resetuje všechny kontrolky, všechny merkery řešící tlačítko STOP (MN8, MN10 – MN15) a setne merker MN9, který informuje o stisku tlačítka inicializace.

### **Řádek 14**

Pokud je napětí v pořádku (Nap\_OK), došlo k stisku tlačítka inicializace (MN9) a není zvolen žádný program, nebo pokud došlo k dokončení jednoho z programů (MN2) dojde ke spuštění inicializace. Jakmile se inicializace dokončí, dojde k resetu informace o tom, že byl dokončen program (MN2), k resetu informace o stisku tlačítka inicializace a k setnutí informace o stavu inicializace Init\_OK.

### **Řádek 15**

Pokud je inicializace hotová, dojde k rozsvícení kontrolky inicializace.

### **Řádek 16**

Pokud není model ve stavu dokončení inicializace, dojde k zhasnutí kontrolky inicializace.

### **Řádek 17**

Stiskem tlačítka TL2 (START) dojde k setnutí merkeru MN16. Toto slouží pro zapnutí napájení po připojení modelu do napájení.

### **Řádek 18**

Tento řádek slouží ke spouštění programů pomocí tlačítka START. Pokud je inicializace hotová a není model zastaven tlačítkem STOP, tak při stisku tlačítka TL2 (START) dojde k setnutí merkeru MN1 a k resetu Init\_OK.

### **Řádek 19**

Tlačítko TL4 odpovídá programu 1. Při jeho stisku za předpokladu, že není model ve stavu STOP, dojde k rozsvícení kontrolky HL3 odpovídající zvolení programu 1 a zároveň k setnutí MN3.

### **Řádek 20**

Tlačítko TL5 odpovídá programu 2. Při jeho stisku za předpokladu, že není model ve stavu STOP, dojde k rozsvícení kontrolky HL4 odpovídající zvolení programu 2 a zároveň k setnutí MN4.

### **Řádek 21**

Tlačítko TL6 odpovídá programu 3. Při jeho stisku za předpokladu, že není model ve stavu STOP, dojde k rozsvícení kontrolky HL5 odpovídající zvolení programu 3 a zároveň k setnutí MN5.

### **Řádek 22**

Tlačítko TL7 odpovídá programu 4. Při jeho stisku za předpokladu, že není model ve stavu STOP, dojde k rozsvícení kontrolky HL6 odpovídající zvolení programu 4 a zároveň k setnutí MN6.

### **Řádek 23**

Tlačítko TL8 odpovídá programu 5. Při jeho stisku za předpokladu, že není model ve stavu STOP, dojde k rozsvícení kontrolky HL7 odpovídající zvolení programu 5 a zároveň k setnutí MN7.

### **Řádek 24**

Za předpokladu, že je Nap\_OK, MN1 informující o stisku tlačítka START a MN3 informující o volbě programu 1 v log. 1, a zároveň informace o volbě ostatních programů (MN4 – MN7) v log. 0, dojde ke spuštění programu 1. Po jeho dokončení se setne MN2 informující o dokončení programu, a resetuje se MN3, MN1 a HL3.

### **Řádek 25**

Za předpokladu, že je Nap\_OK, MN1 informující o stisku tlačítka START a MN4 informující o volbě programu 2 v log. 1, a zároveň informace o volbě ostatních programů (MN3, MN5 – MN7) v log. 0, dojde ke spuštění programu 2. Po jeho dokončení se setne MN2 informující o dokončení programu, a resetuje se MN4, MN1 a HL4.

### **Řádek 26**

Za předpokladu, že je Nap\_OK, MN1 informující o stisku tlačítka START a MN5 informující o volbě programu 3 v log. 1, a zároveň informace o volbě ostatních programů (MN3, MN4, MN6, MN7) v log. 0, dojde ke spuštění programu 3. Po jeho dokončení se setne MN2 informující o dokončení programu, a resetuje se MN5, MN1 a HL5.

### **Řádek 27**

Za předpokladu, že je Nap\_OK, MN1 informující o stisku tlačítka START a MN6 informující o volbě programu 4 v log. 1, a zároveň informace o volbě ostatních programů (MN3 – MN5, MN7) v log. 0, dojde ke spuštění programu 4. Po jeho dokončení se setne MN2 informující o dokončení programu, a resetuje se MN6, MN1 a HL6.

### **Řádek 28**

Za předpokladu, že je Nap\_OK, MN1 informující o stisku tlačítka START a MN7 informující o volbě programu 5 v log. 1, a zároveň informace o volbě ostatních programů (MN3 – MN6) v log. 0, dojde ke spuštění programu 5. Po jeho dokončení se setne MN2 informující o dokončení programu, a resetuje se MN7, MN1 a HL7.

#### **5.3.5.3 Napajeni (Příloha 4)**

### **Řádek 1**

Pokud je vstupní signál 1, dojde nejprve k zresetování chybných stavů os a potom k setnutí merkeru NAP1

### **Řádek 2**

Pokud je merker Err v log. 1, dojde k resetu merkeru NAP1.



### **Řádek 3**

Pokud je NAP1 a vstupní signál v log. 1, dojde k povolení osy AXIS1. O stavu potom informuje merker M12.

### **Řádek 4**

Pokud je NAP1 a vstupní signál v log. 1, dojde k povolení osy AXIS2. O stavu potom informuje merker M13.

### **Řádek 5**

Pokud je NAP1 a vstupní signál v log. 1, dojde k povolení osy AXIS3. O stavu potom informuje merker M14.

### **Řádek 6**

Pokud je NAP1 a vstupní signál v log. 1, dojde k povolení osy AXIS4. O stavu potom informuje merker M15.

### **Řádek 7**

Pokud je vstupní signál v log. 1 a všechny motory jsou povoleny, výstup má hodnotu log. 1.

## **5.3.5.4 Inicializace (Příloha 5)**

### **Řádek 1**

Vstupní signál setne merker M8.

### **Řádek 2**

Merker M8 roztočí osu AXIS1 v kladném směru, tudíž se motor 1 bude pohybovat ke svému referenčnímu snímači. Merker M6 informuje o pohybu této osy.

### **Řádek 3**

Merker M8 roztočí osu AXIS2 v záporném směru, tudíž se motor 2 bude pohybovat ke svému referenčnímu snímači. Merker M7 informuje o pohybu této osy.

### **Řádek 4**

Merker M8 povolí funkci porovnávání, která porovnává aktuální hodnotu referenčního snímače (SQ1) s hodnotou, kterou má snímač v případě, že je aktivní (57). V případě rovnosti dojde k setu M1.

### **Řádek 5**

Merker M8 povolí funkci porovnávání, která porovnává aktuální hodnotu referenčního snímače (SQ2) s hodnotou, kterou má snímač v případě, že je aktivní (58). V případě rovnosti dojde k setu M2.

### **Řádek 6**

M1 informuje o tom, že je AXIS1 na svém referenčním snímači, dojde tak k jeho zastavení pomocí MC\_Halt. Dále pak dojde k nastavení jeho polohy na 0. Potom dojde k setnutí M3.

### **Řádek 7**

M2 informuje o tom, že je AXIS2 na svém referenčním snímači, dojde tak k jeho zastavení pomocí MC\_Halt. Dále pak dojde k nastavení jeho polohy na 0. Potom dojde k setnutí M4.

### **Řádek 8**

Merker M3 spustí blok MC\_MoveModulo, jenž nastaví AXIS3 do pozice 15.8°, setne se M5.

### **Řádek 9**

Merker M4 spustí blok MC\_MoveModulo, jenž nastaví AXIS4 do pozice 0°, setne se M9.

### **Řádek 10**

Pokud je M5 a M9 v log. 1, je inicializace hotová a dojde k setnutí výstupu Out.

### **Řádek 11**

Init\_OK (dokončení inicializace) nebo MN8 (tlačítko STOP) dojde k zresetování všech merkerů používaných v této funkci a k resetu výstupu Out.

## **5.3.5.5 Prog1 (Příloha 6)**

### **Řádek 1**

Pokud je na vstupu tohoto funkčního bloku log. 1, dojde k setu merkeru M23.

### **Řádek 2**

Jestli je tedy M23 v log. 1, dojde k nastavení poměru, v jakém se bude točit slave osa, v našem případě AXIS4, vůči master ose (AXIS3). Poměr je tedy nastaven na -1. To

zajistí, že se bude točit slave stejně rychle jako master, akorát v opačném směru. Po dokončení dojde k setu M16.

### **Řádek 3**

Pokud je M16 v log. 1, dojde k roztočení osy AXIS3 na rychlost 500°/min. Pokud není uveden směr otáčení „Direction“, otáčí se osa automaticky v kladném směru. Zároveň se začne otáčet osa AXIS4 stejnou rychlostí ale v opačném směru, jelikož má nastavený poměr vůči ose AXIS3. Po dosažení daných otáček dojde k setu M17 a M24.

### **Řádek 4**

Je-li M17 v log. 1, dojde k rozjetí osy AXIS1 nastavenou rychlostí na absolutní pozici -89. Po dokončení dojde k setu M18.

### **Řádek 5**

Je-li M24 v log. 1, dojde k rozjetí osy AXIS2 nastavenou rychlostí na absolutní pozici 89. Po dokončení dojde k setu M19.

### **Řádek 6**

Je-li M18 a M19 v log. 1, tedy osy AXIS1 a AXIS2 dosáhly svých cílů, dojde po 2 sekundách k odjetí AXIS1 na pozici -30. Po dokončení se setne M20.

### **Řádek 7**

Je-li M18 a M19 v log. 1, tedy osy AXIS1 a AXIS2 dosáhly svých cílů, dojde po 2 sekundách k odjetí AXIS2 na pozici 30. Po dokončení se setne M21.

### **Řádek 8**

Jsou-li AXIS1 a AXIS2 na svých pozicích zadaných na předchozích řádcích, tedy M20 a M21 jsou v log. 1, dojde k zastavení osy AXIS3, tím pádem i osy AXIS4 jakožto slave zařízení. Následně dojde ke zrušení spojení a převodového poměru mezi AXIS3 a AXIS4 pomocí bloku MC\_GearOut. Po splnění dojde k setu M22.

### **Řádek 9**

Zde je v podstatě konec programu, dojde k setu výstupu „Out“ tohoto funkčního bloku. Zároveň dojde k resetu všech použitých merkerů.

### **Řádek 10**

Merker MN2 v log. 1 resetuje výstup tohoto funkčního bloku.

### **Řádek 11**

Merker MN10 znamenající stisknutí tlačítka STOP resetuje výstup a všechny merkery použité v tomto funkčním bloku.

### **5.3.5.6 Prog4 (Příloha 9)**

#### **Řádek 1**

Pokud je na vstupu tohoto funkčního bloku log. 1, dojde k nastavení poměru, v jakém se bude točit slave osa, v našem případě AXIS4, vůči master ose (AXIS3). Poměr je tedy nastaven na -1. Po dokončení dojde k setu M49.

#### **Řádek 2**

Pokud je M49 v log. 1, dojde k rozjetí osy AXIS1 nastavenou rychlostí na absolutní pozici -89. Po dokončení dojde k setu M50.

#### **Řádek 3**

Je-li M49 v log. 1, dojde k rozjetí osy AXIS2 nastavenou rychlostí na absolutní pozici 89. Po dokončení dojde k setu M51.

#### **Řádek 4**

Je-li M50 a M51 v log. 1, dojde k roztočení AXIS3 na 2500°/min. Po dosažení daných otáček dojde k setu M52.

#### **Řádek 5**

Je-li M52 v log. 1, zapne se časovač, který nám po 3 sekundách setne M53.

#### **Řádek 6**

Je-li M53 v log. 1, dojde k rozjetí osy AXIS1 nastavenou rychlostí na absolutní pozici -30. Po dokončení dojde k setu M54.

#### **Řádek 7**

Je-li M53 v log. 1, dojde k rozjetí osy AXIS2 nastavenou rychlostí na absolutní pozici 30. Po dokončení dojde k setu M55.

#### **Řádek 8**

Je-li M54 a M55 v log. 1, dojde k zastavení rotace ozubených kol a po 1 sekundě dojde k setu M56

### **Řádek 9**

Je-li M56 v log. 1, dojde k roztočení ozubených kol na rychlost 15000°/min. Po dosažení těchto otáček dojde k setu M57.

### **Řádek 10**

Je-li M57 v log. 1, pojede osa AXIS1 na pozici -20. Po dosažení této pozice dojde k setu M58

### **Řádek 11**

Je-li M57 v log. 1, rozjede se osa AXIS2 na pozici 158. Po dosažení této pozice dojde k setu M59.

### **Řádek 12**

Je-li M58 a M59 v log. 1, dojde po 2 sekundách k zastavení rotace ozubených kol. Jakmile se zastaví, dojde k setu M60.

### **Řádek 13**

Je-li M60 v log. 1, dojde ke spojení os AXIS1 a AXIS2 s převodovým poměrem 1. Poté dojde k setu M61.

### **Řádek 14**

Je-li M61 v log. 1, dojde k pohybu osy AXIS1 na pozici -120. Po dosažení této pozice dojde k setu M62.

### **Řádek 15**

Je-li M61 v log. 1, dojde k zapnutí časovače a po 3 sekundách dojde k setu M63.

### **Řádek 16**

Je-li M63 v log. 1, dojde k roztočení ozubených kol na rychlost 5000°/min. Po dosažení této rychlosti dojde k setu M64.

### **Řádek 17**

Je-li M62 a M64 v log. 1, dojde k zastavení rotace ozubených kol. Setne se M65.

### **Řádek 18**

Je-li M65 v log. 1, dojde k odpojení osy AXIS2 od osy AXIS1. Poté pojede osa AXIS1 na pozici -30. Po dosažení této pozice dojde k setu M66.

### **Řádek 19**

Je-li M66 v log. 1, dojde k odpojení osy AXIS4 od AXIS3. Po odpojení dojde k setu výstupu „Out“ funkčního bloku a k resetování všech použitých merkerů v tomto bloku.

### **Řádek 20**

Merker MN2 v log. 1 resetuje výstup tohoto funkčního bloku.

### **Řádek 21**

Merker MN10 znamenající stisknutí tlačítka STOP resetuje výstup a všechny merkery použité v tomto funkčním bloku.

## **5.4 Program pro výuku**

Dále jsem vytvořil program, který by měl sloužit pro výuku na modelu synchronních os.

Jeho základem jsou soubory „Zaklad\_programovani.pro“ a „Zaklad\_programovani.tsm“, které jsou umístěny v příloze 12. Jsou zde zároveň i všechny ostatní soubory, které jsou vytvořeny společně s projektem.

Soubor pro System Manager je v podstatě hotová celá konfigurace, se kterou je možné bez dalších nastavování si vyzkoušet ruční ovládání jednotlivých motorů přes System Manager. Dále je tento soubor potřeba pro spuštění připraveného souboru pro PLC Control.

Soubor pro PLC Control je připraven ve stavu, kdy je hotová celá inicializace. Je zde připraven funkční blok Prog1, ve kterém je připraveno pár řádků programu v jazyce ladder diagram. Řádky 2 – 8 jsou připraveny na vložení bloků jako MC\_MoveVelocity, MC\_MoveAbsolute, MC\_Halt apod. Ty se vkládají mezi kontakt a cívku. Vložení těchto bloků je pak možné docílit sekvence pohybů modelu. Pokud by připravený počet řádků byl nedostačující, je možné další řádky přidat za řádek 8 a pokračovat s dalšími merkery v pořadí. Nyní je poslední merker M23, tudíž přidáním dalšího řádku by na kontaktu byl M23 a na cívce M24. Pokud dojde na rozšiřování, je potřeba poslední použitý merker na cívce dát na kontakt, za kterým následují bloky MC\_Halt a MC\_GearOut. Zároveň je potřeba přidat resetování všech přidáných merkerů na konci programu, což je řádek, kde je na kontaktu X a také přidat resetování všech přidáných merkerů na poslední řádek, který se spouští jen v případě stisku tlačítka STOP.

## Ovládání programu

Tento soubor je připraven pro ovládání přes ovládací panel. Funkce tlačítek STOP, START a INIT je nezměněná, fungují stejně jako v ukázkovém programu. Tlačítko PROG1 je potom připraveno pro napsání vlastního programu. Ostatní tlačítka jsou z programu vyřazena. Pokud by bylo požadováno ovládat místo přes tlačítka celý program přes merkery, jsou připraveny merkery MT1 – MT4, které jsou v programu zařazeny paralelně k tlačítkům TL1 – TL4.

## Informace k programování

Pro roztočení ozubených kol stačí blok MC\_MoveVelocity, použitý pouze pro AXIS3. AXIS4 je již spojen s AXIS3, nepotřebuje tak žádné příkazy. Pro pohyb AXIS1 a AXIS2 by se měl použít blok MC\_MoveAbsolute. Hodnota „Position“ pro AXIS1 by měla být vždy záporná, zatímco pro AXIS2 vždy kladná. Součet hodnot nastavených v „Position“ u AXIS1 a AXIS2 by měl být vždy maximálně 174. Při této hodnotě dojde ke splnutí zubů. Pokud si tedy nastavím pro AXIS1 pozici na -87 a pro AXIS2 pozici 87, dosáhl jsem tak hodnoty již zmiňovaných 174. V případě požadovaného spojení motorů lineárního pohybu by měla být AXIS1 Master a AXIS2 Slave.

Pokud by byl požadován jiný programovací jazyk pro vytvoření pohybu modelu, je možné nahradit Prog1 v MAINu za vámi vytvořený funkční blok. Je však nutné, aby po přivedení signálu do vašeho funkčního bloku došlo hned k použití bloku MC\_GearIn a osa AXIS3 byla Master, osa AXIS4 Slave a převodový poměr byl -1. Na konci programu by mělo zase dojít prvně k zastavení osy AXIS3 pomocí MC\_Halt a následně k odpojení osy AXIS4 od AXIS3 pomocí bloku MC\_GearOut. Následně by měla být možnost resetovat všechny stavy popř. merkery pomocí merkeru MN10 a pomocí MN2 by mělo dojít k resetu výstupu funkčního bloku. V podstatě by se měl přepsat připravený funkční blok v ladder diagramu do jiného jazyka.

## Laboratorní cvičení

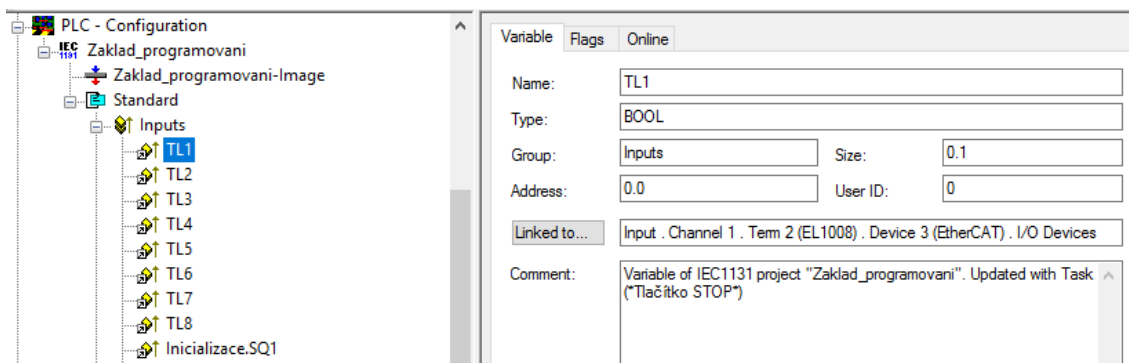
Postup, jak by se mělo postupovat v laboratorním cvičení s použitím připraveného programu.

Pokud je již programové prostředí TwinCAT nainstalováno na PC používaném při laboratorním cvičení, můžeme přejít rovnou k zprovoznění modelu. Spustíme si prostředí TwinCAT System Manager. Otevřeme si připravenou konfiguraci „Zaklad\_programovani.tsm“. Pokud máme správně nastavenou IP adresu v PC, můžeme zkusit aktivovat konfiguraci pomocí příslušného tlačítka. Všechny postupy jsou uvedeny v kapitole „5 TwinCAT a práce s ním – 5.1.1 Základní konfigurace“. V případě, že se povede daná konfigurace aktivovat, měl by se sám systém nastavit do režimu Run Mode. Pokud se nepovede navázat komunikaci, je

potřeba se pokusit vyhledat a připojit k IPC v záložce „SYSTEM – Configuration“. Pokud se navázání komunikace povedlo, lze přejít k otestování pomocí manuálního ovládání os. Zvolíme si jednu z os AXIS3 nebo AXIS4. Jedná se o osy s ozubenými koly. Nemůže tak dojít k lineárnímu pohybu a k případnému poškození modelu. Dále postupujeme podle kapitoly „5.2 Manuální ovládání os“.

Pokud se nám povedlo rozpohybovat některou z os, můžeme si buď vyzkoušet další možnosti manuálního pohybu, nebo přejít k programování.

V záložce „PLC – Configuration“ již máme připojen projekt „Zaklad\_programovani.tpy“. Jelikož uvedená cesta v kolonce „Path“ je nejspíše nesprávná, klikneme na tlačítko „Change...“ a najdeme tento soubor v jeho správném umístění a otevřeme ho. Jelikož se jedná pouze o změnu umístění, nedojde ke smazání nalinkovaných proměnných. To si můžeme ověřit prohlédnutím jednotlivých vstupů a výstupů viz Obr. 64. U každého je šipka, což znamená, že je nalinkován. Popř. kliknutím na ně se nám otevře v pravé části prostředí okno, kde lze vidět vedle tlačítka „Linked to...“ místo, kam je daná proměnná nalinkována. Pokud zde nic není, tedy není nalinkována, musíme provést linkování.



**Obr. 64 - Ověření linkovaných proměnných**

Pokud tedy nejsou nalinkované proměnné, je potřeba je nalinkovat podle následující tabulky (Tab. 5). Postup správného linkování je uveden v kapitole „5.3.3.5 Linkování proměnných v System Manageru“.

**Tab. 5 - Linkování proměnných**

Inputs	Linked to	Outputs	Linked to
TL1	Term2 – 0.0	HL1	Term4 – 0.0
TL2	Term2 – 0.1	HL2	Term4 – 0.1
TL3	Term2 – 0.2	HL3	Term4 – 0.2
TL4	Term2 – 0.3	HL4	Term4 – 0.3
TL5	Term2 – 0.4	HL5	Term4 – 0.4
TL6	Term2 – 0.5	HL6	Term4 – 0.5
TL7	Term2 – 0.6	HL7	Term4 – 0.6



TL8	Term2 – 0.7		
SQ1	Drive23 - _IO_LIO_act		
SQ2	Drive24 - _IO_LIO_act		
S1	Drive23 - State		
S2	Drive24 - State		
S3	Drive26 - State		
S4	Drive27 - State		

Proměnné AXIS1 – AXIS4 se linkují na stejně číslované osy Axis1 – Axis4.

Pokud je však vše nalinkováno, nemusíme tuto záležitost řešit a můžeme přejít rovnou k připravenému programu.

Otevřeme si prostředí TwinCAT PLC Control. V něm si otevřeme připravený program „Zaklad\_programovani.pro“. Lze vidět, že jsou zde již připravené jisté funkční bloky i hlavní program Main. Jak postupovat s programem je popsáno v podkapitole „5.4 Program pro výuku“. Pokud se připravený program nahraje do IPC v neupraveném stavu, bude fungovat pouze inicializace a tlačítka STOP a START. Program Prog1 je potřeba doprogramovat. Sled pohybů je již na řešiteli.

Postup nahrání programu je popsán v podkapitole „5.3.4 PLC Control online režim“.

## 6 ZÁVĚR

V první části této práce jsem se seznámil s celým modelem synchronních os. Dále jsem se potom seznámil i s prostředím TwinCAT.

Byla předělána komunikace motorů s IPC. Nyní jsou tedy motory připojeny do karet EK1122. S touto předělkou byl přidán i jednoduchý ovládací panel, který slouží k volbě jednoho z pěti programů vytvořených pro ukázkou funkčnosti tohoto modelu.

Nejdůležitějším prvkem pro roztočení motorů je přidání těmto motorům startup parametry, které zajistí správný chod po připojení modelu do elektrické sítě a nahrání parametrů motorů. Správnost nastavení jsem si potom ověřil pomocí manuálního ovládání os.

Po provedení této části jsem pak už mohl přejít k vytváření programu v PLC Control. Co se týče motorů lineárního pohybu, musel jsem zjistit, který signál hlídá najetí na referenční snímač. To z důvodu, že tyto snímače jsou připojeny do signálových I/O, které jsou v elektronice motorů. Motory s ozubenými koly si vždy na začátku programu, tedy v inicializační části, natočím do pozice takové, aby bylo možné s nimi do sebe zajet. Ozubené kolo má 13 zubů a 13 mezer. Kdyby v nulové pozici byly zuby přesně naproti sobě, bylo by nutné jeden z motorů pootočit o  $13,8^\circ$ . To je maximální hodnota, o kterou by se musel jeden z motorů pootočit na jednu nebo druhou stranu pro přesné zapadnutí zubu jednoho kola do mezery druhého kola. Jelikož vytištěná ozubená kola do sebe v nulové pozici nezapadnou tak, jako polystyrenová, vybral jsem si osu AXIS3, se kterou budu pootáčet a osa AXIS4 je natočena do nulové pozice. V této kombinaci je potřeba osu AXIS3 otočit do kladného směru o  $15,8^\circ$ . Po inicializaci je tak možné již spustit některý z programů, kde je první věc řešeno vždy spojení os s ozubenými koly pomocí bloku MC\_GearIn.

Vytvořil jsem tak 5 ukázkových programů. Ty je možné jednotlivě spouštět pomocí ovládacího panelu. Na konci každého programu se provede inicializace sama. Nejdůležitějším prvkem programu je inicializace, která nastaví ozubená kola do pozice připravené ke splynutí ozubených kol. Část programu zajišťující zastavení modelu tlačítkem STOP jednak zastaví motory lineárního pohybu pomocí MC\_Stop, a také zastaví motory s ozubenými koly. Ty však pouze pomocí MC\_Halt a to pro případ, že by byly zrovna spojené blokem MC\_GearIn. To zajistí bezpečné zastavení bez poškození ozubených kol.

Připravil jsem ještě jeden program, který by měl sloužit pro výuku na tomto modelu. Program pro výuku je založen na stejném principu jako program ukázkový s tím rozdílem, že dostupný je pouze program Prog1, ve kterém si uživatel sám naprogramuje posloupnost pohybů modelu. Studenti na pracovišti tak potřebují PC s programovým prostředím TwinCAT a soubory připravené pro výuku. Jedná se o

kompletně hotovou konfiguraci v prostředí TwinCAT System Manager a o připravený program do prostředí PLC Control. Studenti by měli dokázat s domácí přípravou v dvouhodinové výuce rozpohybovat model. Dopsání programu tak, aby došlo ke splynutí ozubených kol, by se studenty demonstrativně provedl vyučující. Následně by studenti přidali do programu další prvky pohybu.

# Literatura

- [1] VŠPJ: Fotogalerie [online]. [cit. 2017-01-03]. Dostupné z:  
<https://fotogalerie.vspj.cz/fotogalerie-skola>
- [2] C6915-0000 | Fanless control cabinet Industrial PC. Beckhoff [online]. [cit. 2017-01-03]. Dostupné z:  
[https://www.beckhoff.com/english.asp?industrial\\_pc/c6915\\_0000.htm](https://www.beckhoff.com/english.asp?industrial_pc/c6915_0000.htm)
- [3] EK1100 | EtherCAT Coupler [online]. [cit. 2017-01-03]. Dostupné z:  
<https://www.beckhoff.com/english.asp?ethercat/ek1100.htm>
- [4] EK110x, EK15xx. : EtherCAT Bus Coupler [online]. 2016, s. 79 [cit. 2017-01-03]. Dostupné z:  
[https://download.beckhoff.com/download/document/io/ethercat-terminals/ek110x\\_ek15xxen.pdf](https://download.beckhoff.com/download/document/io/ethercat-terminals/ek110x_ek15xxen.pdf)
- [5] EtherCAT Terminals. Beckhoff [online]. [cit. 2017-01-03]. Dostupné z:  
[https://www.beckhoff.com/english.asp?ethercat/ethercat\\_terminals.htm?id=235616333](https://www.beckhoff.com/english.asp?ethercat/ethercat_terminals.htm?id=235616333)
- [6] BARTOŠÍK, Petr. Automa [online]. 2015, 2015(06) [cit. 2017-01-03]. Dostupné z: [http://automa.cz/cz/casopis-clanky/twincat-3-extended-automation-technology-xat-2015\\_06\\_53801\\_6979/](http://automa.cz/cz/casopis-clanky/twincat-3-extended-automation-technology-xat-2015_06_53801_6979/)
- [7] Lexium Integrated Drive [online]. 2012 [cit. 2017-01-03]. Dostupné z:  
[http://download.schneider-electric.com/files?p\\_File\\_Id=4218285553&p\\_File\\_Name=ILA2E\\_EtherCAT\\_Manual\\_V202\\_EN.pdf](http://download.schneider-electric.com/files?p_File_Id=4218285553&p_File_Name=ILA2E_EtherCAT_Manual_V202_EN.pdf)
- [8] Beckhoff Information System [online]. [cit. 2017-01-03]. Dostupné z:  
[https://infosys.beckhoff.com/english.php?content=../content/1033/html/bkinfosys\\_intro.htm&id=7149523117184201825](https://infosys.beckhoff.com/english.php?content=../content/1033/html/bkinfosys_intro.htm&id=7149523117184201825)
- [9] Beckhoff Automation [online]. [cit. 2017-01-03]. Dostupné z:  
<https://www.beckhoff.be/english.asp?beckhoff/default.htm?id=3536>
- [10] VOJÁČEK, Antonín. Programovací režimy pro PLC dle IEC 61131-3 (CoDeSys) [online]. 2011 [cit. 2017-01-03]. Dostupné z:  
<http://automatizace.hw.cz/programovaci-rezimy-pro-plc-dle-iec-611313-codesys>
- [11] EK1122 | 2-port EtherCAT junction [online]. [cit. 2017-05-02]. Dostupné z:  
<https://www.beckhoff.com/english.asp?ethercat/ek1122.htm>

- [12] ZS1090-0003 [online]. [cit. 2017-05-02]. Dostupné z:  
<https://download.beckhoff.com/download/document/io/accessories/zs1090-0003.pdf>
- [13] TinkerCAD [online]. [cit. 2017-05-02]. Dostupné z:  
<https://www.tinkercad.com/>
- [14] MOLEX 430300007. TME [online]. [cit. 2017-05-05]. Dostupné z:  
<http://www.tme.eu/cz/details/mx-43030-0007/signalove-konektory-raster-300mm/molex/430300007/>
- [15] MOLEX 430251200. TME [online]. [cit. 2017-05-05]. Dostupné z:  
<http://www.tme.eu/cz/details/mx-43025-1200/signalove-konektory-raster-300mm/molex/430251200/>
- [16] ZEŽULKA, František a Ondřej HYNČICA. Průmyslový Ethernet IX: EtherNet/IP, EtherCAT. *AUTOMA*. 2008, 2008(10), 3-4.

## Seznam příloh

Příloha 1. Návrh jednotlivých dílů krabičky pro ovládací panel

Příloha 2. Globální proměnné

Příloha 3. Main

Příloha 4. Napajeni

Příloha 5. Inicializace

Příloha 6. Prog1

Příloha 7. Prog2

Příloha 8. Prog3

Příloha 9. Prog4

Příloha 10. Prog5

Příloha 11. Ukázkový program

Příloha 12. Program pro výuku